Department of Electrical & Computer Engineering
Software Engineering
The University of Auckland
New Zealand

Transparency in Software Engineering

Yu-Cheng Tu

May 2014

Supervisors: Professor Clark Thomborson
Associate Professor Ewan Tempero

A THESIS SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS OF DOCTOR OF PHILOSOPHY IN ELECTRICAL AND ELECTRONIC ENGINEERING

Abstract

Transparency has the meaning of making information visible to people. Good transparency enhances the reputation of organisations and enables people to make informed decisions. Transparency is widely used in software engineering, but it is unclear how the concept of transparency might help software development. Two questions inspire this thesis: What is transparency in software engineering? How useful is transparency to software development?

Current definitions in software development lack specific measurable characteristics and ways to measure transparency. We propose an introductory definition of transparency as it relates to software development: the degree to which stakeholders can answer their questions by using the information they obtain about a software system during its life cycle. This definition rests on three attributes: accessibility, understandability, and relevance. These attributes affect stakeholder's ability to see the information necessary to achieve their goals.

We use evidence from an exploratory survey which asks software practitioners their personal opinions about transparency. We also collect evidence from a controlled experiment which compares two software artefacts with different degrees of transparency in presenting functional requirements to software practitioners and tertiary students. This experiment enables us to test how useful transparency is in requirements engineering.

The results from the exploratory survey reveal that software practitioners encounter transparency problems during communication in their software projects. The findings from the controlled experiment illustrate that a more transparent software artefact is more effective in presenting functional requirements than a less transparent software artefact. Future research in this area will give software developers a diagnostic framework which enables them to articulate transparency problems and to improve communication in the software life cycle.

Acknowledgements

Firstly, I would like to thank my supervisors Professor Clark Thomborson and Associate Professor Ewan Tempero, for their support and guidance during the time of this research. They always provided interesting ideas and useful advice. I always learned something interesting from our weekly meetings. I am very grateful for their encouragement throughout this research.

I would like to thank Dr Catherine Watson, for her feedback about my research. I would also like to thank her for the administration of this research in the Department of Electrical and Computer Engineering.

I would like to thank the anonymous organisation for providing the requirements document for the experiment. I would also like to thank anonymous software engineers for their helpful feedback about "transparency". Moreover, I would like to thank all participants who took part in the research.

I would like to thank Pita Jarupunphol, Habib Naderi, Moon-Ting Su, Steven Hu, Se-Young Yu, Maziar Janbeglou, Marc Jeanmougin, and many friends from the Department of Computer Science for their friendship and support. We always have interesting discussions about various aspects of our research. Moreover, thank you to Hanlie Van Zyl and Christine Salter from the Department of Electrical and Computer Engineering as well as Robyn Young and Sithra Sukumaar from the Department of Computer Science for providing me with all the necessary help during the research.

I would also like to thank Barbara Thomborson for proofreading and editing the grammar of this thesis. I am very grateful for her suggestions for improvements of this thesis.

Finally, I would like to thank my family for providing me support and for coping with me throughout the time of this research. I am very grateful for their support and encouragement.

Contents

THE UNIVERSITY OF AUCKLAND
SCHOOL OF GRADUATE STUDIES

Co-Authorship Form

Graduate Centre
ClockTower – East Wing
22 Princes Street, Auckland
Phone: +64 9 373 7599 ext 81321
Fax: +64 9 373 7610
Email: postgraduate@auckland.ac.nz
www.postgrad.auckland.ac.nz

This form is to accompany the submission of any PhD that contains research reported in published or unpublished co-authored work. **Please include one copy of this form for each co-authored work.** Completed forms should be included in all copies of your thesis submitted for examination and library deposit (including digital deposit), following your thesis Acknowledgements.

Please indicate the chapter/section/pages of this thesis that are extracted from a co-authored work and give the title and publication details or details of submission of the co-authored work.

Title: Illusions and Perceptions of Transparency in Software Engineering. Published in the 18th Asia-Pacific Software Engineering Conference (APSEC), 2011.

Parts of this publication are reported in Chapter 5/section 5.1 Survey Design/pages 50 – 54, section 5.2 Execution/pages 56 – 57, section 5.3.4 Transparency and Communication Problems/page 75, section 5.4 Threats to Validity/pages 77 – 79, Chapter 7/section 7.2 Exploring Transparency/pages 151 – 152.

Nature of contribution by PhD candidate: The PhD candidate designed the survey study, analysed the data, wrote the text, and created figures for this publication.

Extent of contribution by PhD candidate (%): 75%

CO-AUTHORS

Name	Nature of Contribution
Clark Thomborson	Improved the statistical analysis, provided suggestions for the interpretation of results for this publication.
Ewan Tempero	Improved the evaluation of the survey, provided suggestions for the structure of this publication.

Certification by Co-Authors

The undersigned hereby certify that:

- ❖ the above statement correctly reflects the nature and extent of the PhD candidate's contribution to this work, and the nature of the contribution of each of the co-authors; and
- ❖ in cases where the PhD candidate was the lead author of the work that the candidate wrote the text.

Name	Signature	Date
Clark Thomborson		5/09/2013
Ewan Tempero	ETempero	5/09/2013
		Click here
		Click here
		Click here

1
Introduction

The term “transparency” appears in many areas with different implications. Organisations emphasise transparency to promote open information and operations. For example, Microsoft’s Open Government solution [75] makes information transparent for government agencies and citizens. Transparency in this example implies that government agencies and citizens can see the government’s information.

Transparency on the other hand implies a process is not easily seen or noticeable, which is useful to distributed computing. A guideline for configuring a transparent process in the DCS (Distributed Connectivity Services) by Microsoft illustrates transparency. The DCS provides an infrastructure and tools for building distributed service solutions [76]. A transparent process in the context of the DCS “executes silently” when an operation is invoked [77].

Transparency also appears in software engineering with different implications. In many software engineering-related areas, transparency generally refers to a product or a development process’s visibility to stakeholders. For example, Scrum, an agile software development methodology, highlights the value of transparency. Transparency in Scrum concerns making a development process visible to observers [97]. In code inspection, transparency is about a user’s ability to look into the source code if he or she encounters a problem [71]. Moreover, according to the Software Engineering Body of Knowledge (SWEBOK), transparency is one of the principles for guiding the SWEBOK project, where

> "the development process is itself documented, published, and publicized so that important decisions and status are visible to all concerned parties" [1].

Similarly, according to Ghezzi et al. [48], transparency or visibility is a quality in which

> "a development process is visible if all of its steps and its current status are documented clearly."

In seeming contradiction, transparency refers to a computational process or artefact is unnoticeable. For example, according to the Oxford Dictionary of the Internet [54], transparency is a property that "makes the user of a network unaware of the fact that they are interacting with a network". Similarly, transparency in distributed systems implies that users cannot distinct access to local resources from access to remote resources [108].

Examining how the term "transparency" is defined in different areas reveals the complexity of the concept of transparency. Different implications of transparency can be useful to various aspects of software engineering. In particular, transparency with the implication of visible information can be useful to improve the software development process. A lack of transparency can hinder communication among stakeholders during software development. For example, project managers might become concerned if the software developers were not transparent about their work. The project managers might be unable to make decisions if they could not be certain that the software developers were working according to the plan. The project managers would need to talk to the software developers again to find information necessary to make decisions. This in turn affects the project progress.

The above example illustrates the project managers' need to know during the software development process. Transparency's implication of visible information is important to project managers because they can see information about the work done by software developers and make decisions based on what they see. Transparency with the paradoxical implication of unnoticeable information is undesirable for project managers in the example as they need information from software developers to make meaningful decisions. Therefore, to satisfy project managers' need to know, it is important for software developers to make information visible to project managers. Project managers can acquire or receive information via some communication channels provided by software developers.

In this thesis, we focus on exploring transparency with the implication of the information about a product, a development process etc., is visible to any stakeholders. The concept of transparency in this thesis generally refers to the meaning of making information visible to stakeholders.

However, there exists very little investigation of how the concept of transparency could aid software development. Current definitions of transparency are inconsistent in the literature. It is unclear what transparency is improving or how transparency is assessed in software engineering. In addition, the term "transparency" or "transparent" appears in much software engineering literature without a proper definition. For example, Paul and Tanenbaum [86] present an approach for a trustworthy electronic voting system which is based on "the use of open source software, transparent procedures, and simple cryptography". Paul and Tanenbaum do not define what transparent procedures are, but the context suggests that people can see the procedures involved in implementing an electronic voting system. Similar examples of the usage of "transparency" can be found in papers on social software engineering, information privacy, and graphical programming [66, 83, 87, 90, 121].

In this thesis, we explore two questions: What is transparency in software engineering? How useful is transparency to software development? We use notions of accessibility, understandability, and relevance as they relate to transparency. These three attributes are important to achieve transparency in software engineering. We believe that if developers explicitly think about the concept of transparency can improve communication with other stakeholders during the software life cycle. The concept of transparency can reduce problems with communicating information in software systems and projects.

1.1 Problems with Communicating Information

A lack of transparency can affect communication of information to stakeholders during software development. Before discussing how transparency affects communication, we first present examples of problems with communicating information in software systems and projects.

In software systems and projects, examples abound of famous failures that led to disastrous consequences. For example, the computer system developed for the London Ambulance Service (LAS) in 1992 failed in less than two days of operation. Problems in the system caused delays in dispatching ambulances. As a consequence of ambulance delays, people died [44].

Failures in software systems have also incurred large financial costs to governments and organisations. To illustrate, the Ariane 5 rocket failure in 1996 cost USD$350 million [18]. A software exception caused the failure, where the rocket broke up and exploded about 40 seconds after initiation of the flight sequence [65].

Similarly, failures in software projects cost governments and organisations greatly. The cost of the INCIS project for the New Zealand Police in the 1990s incurred a cost of over NZD$100 million. The goal of the project was to develop a system that provided criminal information for the police. However, the system was abandoned before completion [32, 103].

Examining failures in software systems and projects reveals many different factors contributing to software project failures. Some of the common factors include unrealistic or unarticulated project goals, badly defined system requirements, poor reporting of the project's status, and poor communication among customers, developers, and users [18]. Verner et al. [118] discuss a similar set of failure factors such as uninvolved stakeholders, vague requirements, and no intra-team communication. In information system failures, Lyytinen and Hirschheim [68] identify 16 failure classes which include technology problems, data problems, complexity problems, and communication problems.

The above examples and research show information about a software system or a software project is important for helping stakeholders achieve their goals. Stakeholders use information to learn how a software system works and to perform tasks on that system. Furthermore, stakeholders use information to manage software projects and to make decisions about the functionality of a software system. However, it is not always easy for stakeholders to acquire or receive the information necessary to achieve their goals.

Often identifying and acquiring the information stakeholders need to achieve their goals is the difficulty. For example, project managers might not be able to find the real project status because people tried to cover bad news about the software project [110]. The LAS system in 1992 faced such a problem as no exception reports about the system existed. To mitigate the risk of covering up bad news, people disclosed problems in the 1996 turnaround project for the LAS system [41]. Other issues also affect stakeholders' ability in acquiring the information that they need. For example, software developers might not be available to answer clients' questions immediately due to disparate locations of developers and clients.

Another problem with information in software systems and projects is the difficulty in understanding information by stakeholders. Non-expert stakeholders might find technical details about a software system difficult to understand. This problem could cause negative feelings about a software system such as stakeholders' feelings of alienation about the software and feelings of discounting their concerns [68]. Information such as business requirements might also be difficult to understand by software developers who are not expert in the problem domain. This could prevent software developers from creating systems that satisfy stakeholders' needs.

Stakeholders could also face problems in identifying the relevant piece of information they need to achieve their goals. Problems occur when there is too much information available or when the information that stakeholders need is stored in disparate locations. These situations might cause stakeholders to overlook important parts relating to their need, which in turn could lead to "inefficient use of decision-making time" [39].

These examples show that stakeholders have problems with accessing, understanding, and identifying information relevant to their interests. These problems are concerned with accessibility, understandability, and relevance of information, which affect the degree of transparency in the information communicated to stakeholders. If information is transparent, stakeholders should easily see the information that they need to answer their questions.

Before we explore what transparency is, we first present an overview of communication in software engineering in the following section. The purpose of the overview is to help readers understand how transparency is related to communication in software engineering. We also present a simple communication model for describing transparency in the context of communication in Section 1.3. In addition, we discuss communication problems using the simple communication model in Section 1.4.

1.2 Communication in Software Engineering

Communication, according to the Oxford Dictionary of English, is

> "the imparting or exchanging of information by speaking, writing, or using some other medium" [106].

Communication is important for developing software systems [118]. It is also an important aspect for user participation in the development of information systems [51]. Communication in software engineering is generally considered as stakeholders communicating with each other directly or through some medium during the development of a software system. For example, according to Hartwick and Barki [51], a communication activity is

> "the performance of information exchange activities as users communicate formally and informally with other participants."

Communication activities occur in different ways in a software project and include face-to-face discussions, group meetings, and formal documentation. The purpose of communication differs depending on the type of stakeholders involved in a communication activity. Different stakeholders are interested in different aspects of a software system. Different types of stakeholders are involved in a software project. Poole [88] divides the stakeholders into two main groups:

- Stakeholders from professional services such as software engineers, project managers, quality assurance engineers, and user interface designers.
- Stakeholders for clients such as project champion, users of a software system to be developed, and marketing directors.

The purpose of communication also differs depending on the phase of a software life cycle. For example, at the initial stage of software development, members of a software design team need to acquire information from sources such as documentation, formal training sessions, and other project members [119]. In the requirements engineering phase, Coughlan and Macredie [24] classify communication activities based on the discussion by Walz et al. [119]. These activities are knowledge acquisition, knowledge negotiation, and user acceptance. This classification suggests that project members in the requirements engineering phase not only need to acquire information for the project, they also need to negotiate requirements with other stakeholders. Project members need to make sure that stakeholders accept the requirements after the negotiation of requirements.

Each communication activity in a software project has a different purpose; these purposes depend on the type of stakeholders involved in the communication activity and the phase of the software life cycle. Three reasons underpin the purpose of communication as Da Silva and Agusti-Cullell [29] describe them in the context of artificial intelligent systems. These three reasons, discussed below, are relevant to agents in the software development project.

1. Communication to explain.

   In a communication to explain, an agent (sender agent) is explaining some aspects of the reality to another agent (receiving agent) who has no direct access to the reality. In Da Silva and Agusti-Cullell's model, agents are people who are "temporally and physically situated". The receiving agent connects with an artificial information system proposed by the sender agent to connect with the reality. An artificial information system or a model is a "simplification of the reality as perceived by the agent who formulates it, based on an information system".

   An example of communication to explain is when a client of a software project is interested in knowing the current status of this project. The client is the receiving agent and the project manager can be the sender agent. The client may use a progress report prepared by the project manager to understand the current project status.

2. Communication to command.

   In a communication to command, an agent has "a goal to reach and looks for instructions to act in order to reach its goal [in] the most effective way". Here, the agent is a receiving agent who receives instructions from a sender agent. The receiving agent attempts to modify some aspects of the reality to achieve his or her goal using the sender agent's artificial information system. The sender agent may be someone who has a similar goal and has recorded information about an artificial information system. The sender agent may also be someone who is the expert in the field of interest of the receiving agent. The expert sender agent has direct access to a part of the reality which is relevant to the receiving agent's goal. Therefore, as explained by Da Silva and Agusti-Cullell, the success of communication to command depends on

   > "how well the reality of interest for the receiving agent matches with the one conveyed in the artificial information system proposed by the sender agent, as well as how precise are the mutual presuppositions of the sender and the receiving agent."

   An example of communication to command in the context of software engineering is when a software developer aims to develop a software system that meets client's requirements. The software developer is the receiving agent for a requirements specification, which is an artificial information system from a requirements engineer (the sender agent). The software developer in this example develops a software system using the requirements specification. The requirements specification should reflect the client's requirements.

3. Communication to satisfy.

   In a communication to satisfy, an agent is attempting to use an artificial information system that conveys a virtual reality. The agent is a receiving agent who uses the artificial information that is created by a sender agent. For example, the receiving agent may be an end user of a software system and the sender agent may be the software developer of the software system. The sender agent should have an appropriate understanding of the receiving agent so that the artificial information system satisfies some desires or needs of the receiving agent. The receiving agent also knows and accepts the sender agent before adopting the artificial information system.

   An example of communication to satisfy in the context of software engineering is when end users accept a software system which is implemented by software developers. The end users then attempt to use the software system to satisfy their needs according to the behaviours of the software system.

Communication in software engineering involves various stakeholders who have different intents. The type of information acquired or received by stakeholders depends on the stakeholders' purpose of communication. The concept of transparency is useful to the sender agent for fulfilling the purpose of communication and is useful to stakeholders regardless of the reasons to communication. If information is not transparent to stakeholders who are receiving agents, they will not understand the information and be unsatisfied. Stakeholders are unable to communicate to explain, to command, or to satisfy when they cannot obtain or understand the artificial information system that the sender agent has prepared. Moreover, communication cannot be successful if the artificial information system is not relevant to stakeholders.

To describe what is involved in communication to achieve transparency for software development, we present a simple communication model in the following section. The following section enables readers to understand basic components of communication that are important to software engineering.

1.3 A Simple Communication Model

A simple communication model helps us to discuss communication problems in terms of the main components of communication. The model helps us to see how transparency is relevant to communication in software development. The simple communication model is based on Shannon's mathematical theory of communication [99]. Figure 1.1 illustrates the simple communication model. It has three main components:

1. Sender.

   A sender does the transmitting in a communication system. The sender can produce a suitable signal and provide a channel for sending messages. The sender can also be the source that originates a message or a sequence of messages. A message is some information that a sender has in his or her mind that the sender wishes to convey to a receiver. A signal is a message translated from the sender's mind into verbal, written, or recorded information. In this thesis, we focus on the signal transmitted in a communication channel. We use the term "information" for referring to signals produced by senders.

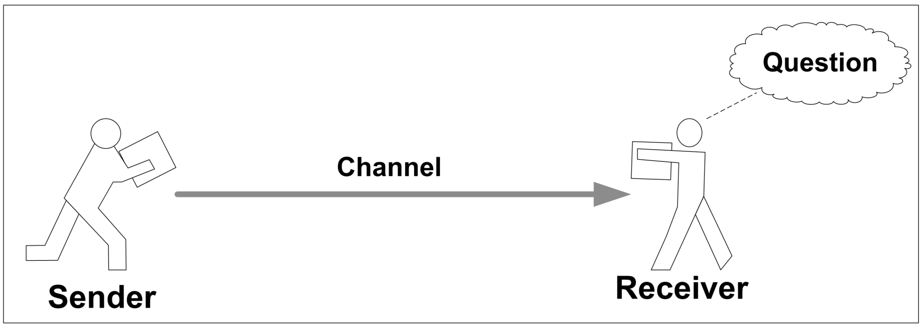


Figure 1.1: A simple communication model.

2. Channel.

   A channel is the medium for a sender to send signals. It is the means for conveying information from a sender to a receiver. Examples of channels are print media, individuals, and electronic files. According to Case [16], "different sources can inhabit one type of channel." For example, if someone reads a printed document, the channel is the document and the sources are the authors of that document.

3. Receiver.

   A receiver receives signals from a sender via a channel. The receiver has a set of questions in his or her mind. The questions concern the purpose of communication between the sender and the receiver. The receiver then reconstructs messages from the signals received to answer his or her questions.

As illustrated in Figure 1.1, communication, in the simplest terms, involves a sender passing information to a receiver via a channel. The receiver seeks answers to their questions. For example, a software developer is developing a software system that meets the client's requirements. This is an example of communication to command as discussed in the previous section. The receiver in this example is the software developer, and the sender is the requirements engineer who is eliciting requirements for a software system. The requirements engineer (sender) produces a requirements specification document which in turn is the communication channel for conveying requirements to the developer. The developer might have questions such as, "What are the functional requirements for the system?" The developer then looks through the document to answer his or her questions.

In some cases, information is not sent directly to the receiver but is available in some repository. The receiver needs to access the repository to retrieve information. This suggests that the receiver will need to look for the repository and request information from the repository. The receiver might then need to wait for the repository to reply. The reply from the repository might become obsolete if the receiver does not receive the reply in time for completing his or her tasks. In the simple communication model, the repository is treated as the sender of information. Unlike the above example, where the information is pushed to the receiver, the receiver tries to pull information from the repository by first sending a request to the repository.

In the following section, we present an overview of problems relevant to communication in software engineering. We also discuss communication problems with respect to the simple communication model.

1.4 Communication Problems in Software Engineering

In this section, we discuss communication problems in software engineering using the simple communication model. The simple communication model involves a sender, a receiver, and a channel to transmit information. To achieve an effective communication, according to Cerri [17], the sender should match the receiver's structure of reality. The sender should also check whether the message received by the listener (receiver) is as intended. An effective communication, as explained by Cerri, is "the ability to communicate so that the listener's [(receiver)] 'filters' are not engaged..." Cerri explains that each person collects data about the world from his or her senses and integrates the data into a "map" of reality. People use their map of reality to make decisions and to filter information from the world. Deletion, distortion, and generalisation of information could hinder the effectiveness of communication as they change what is intended in the communication.

According to Lyytinen and Hirschheim [68], distortion of information is an especial problem in information systems. Information, or data, could distort the true picture of an organisation. This in turn might lead users to take inappropriate actions. Furthermore, information could be incorrect or lack relevance due to wrong classification schemes and measurement categories. These problems hinder users in realising their expectations in the use and the maintenance of information systems. In the simple communication model, distortion of information could occur at the sender side. The sender may send only parts of the true pictures to receivers. The sender could also distort facts by sending misrepresented data to receivers. In addition, the sender could produce information that is inaccurate or irrelevant to the receiver. In such cases, the sender does not intend to distort the information.

Another problem from the sender side is associated with missing or inaccessible data. For example, rationales for the decisions during the requirements engineering phase might not be recorded [2]. Data could become impossible to access in a cost-effective manner [68].

These problems would hamper the receiver from acquiring information from the sender.

Stories of retrieval issues for the receiver are found in studies on how engineers communicate with others. According to Hertzum and Pejtersen [52], quick and easy access of information is important for engineers to choose which channels when they seek information. Hertzum and Pejtersen also discuss barriers that affect engineers' choice of channels for written and verbal information. Example barriers for finding written information include cost in time, irrelevant information, and poor availability of information. Example barriers that affect engineers' choice of verbal information include cost in time and too much effort required to involve the other party. In addition, information might be too general or irrelevant to address engineers' problems. Moreover, the engineer's experience with the source affects the perception of accessibility of an information source. Tenopir and King [111] summarise that the choices of ways engineers communicate depend on

> "the perceived likelihood of success within an acceptable time period and on perception of relative accessibility, cost (i.e., time and expenditure), and effort necessary to obtain the information".

Too much information also affects receivers' ability in finding answers to their questions. This problem can be found in the requirements engineering phase, for example, where programmers have to interpret raw natural language. In the interviews conducted by Al-Rawas and Easterbrook [2], one programmer complained that instead of reading a diagram or formal notation, he had to read a large amount of text to understand a single requirement.

At the other end of the communication model, receivers may misunderstand information, a common problem in software projects. A change in scope, an incomplete description of the project, or a difference of opinion among stakeholders could cause misunderstandings on a project [88].

The communication channel could also affect the receiver's ability to find answers to his or her questions. For example, in large software projects, documentation is one form of communication among individual project members as well as between successive teams [27]. It is a one-way communication channel in the requirements engineering phase [2]. However, documentation is often ineffective for communication as it is difficult to resolve misunderstandings between stakeholders. Moreover, documentation is often late and incomplete. The formats used in documentation might be insufficient for communicating some design information. In some cases, some information might not be recorded because of schedule pressures [27].

Incomprehensible information and the use of unfamiliar language are also problems that affect the receiver's ability in having their questions answered. For example, a communication difficulty during the requirements engineering phase is related to the notations used for requirements specifications. Problems such as misunderstanding of requirements occur when different groups of stakeholders are unfamiliar with the notations used to model the requirements [2]. Al-Rawas and Easterbrook [2] found that 86% of developers commented that their customers would normally need additional explanation in order to understand the notations used to specify requirements. Similarly, communication problems occurred when one party could not understand the terminology used for communicating technical matters during requirements elicitation [93]. These problems suggest that the receiver in the simple communication model could not understand the information received and consequently could not have his or her questions answered.

In summary, there are different types of communication problems in software engineering that affect the receiver's ability in finding answers to his or her questions. The sender and the communication channel affect how well the receiver answers his or her questions. To reduce communication problems, the concept of transparency would be useful for improving the information presented in the communication channel. When the sender communicates to the receiver with the concept of transparency in mind or utilises the concept of transparency during communication, the receiver can access and understand the information that they need to answer his or her questions.

1.5 An Overview of "Transparency"

Transparency is a term that appears in various contexts such as business, computing, and public participation. The definition of transparency differs depending on the context. A quick search in Oxford Reference Online returns 42 results relating to "transparency". These results contain definitions as well as subject references in science, philosophy, business, law, and computing. For example, the first definition of transparency, according to the Oxford Dictionary of English, is "the quality or condition of being transparent; perviousness to light; diaphaneity, pellucidity." [106].

Transparency as a physical object refers to a piece of photographic film used for projecting pictures. It is also the quality of an object such as glass which can be seen through. Transparency in science refers to the degree to which a medium allows radiation to pass through [4, 35]. Transparency in business and law has the notion of information openness, which suggests that information in a transparent document or process is available and visible [15, 62, 78]. On the other hand, transparency in distributed computing suggests that users are unaware of the computational process or artefacts [54, 108].

Transparency is an important principle for organisations, particularly for governments and large corporations. Transparency influences the success, reputation and credibility of organisations [84]. Transparency is also a criterion for evaluating the effectiveness of public participation [10, 91]. Organisational transparency connotes information openness, which is important for making information about governments and organisations visible to the public. It enables the public to see the outcome of public participation.

Transparency is also an ethical principle for organisations. Discussions of ethics and transparency are found in a special issue of Ethics and Information Technology [116]. Transparency is important for enhancing public acceptance and for demonstrating fairness of organisations in decision-making. Transparency implies the quality of a process, statement, or information being easily understood or recognised.

In software engineering, the term "transparency" generally refers to the notion of information being visible or open to stakeholders. It is now an important concept in software development. This notion of transparency helps stakeholders to make decisions based on the information disclosed [11, 48]. It is also a virtue for providing assurance to people and for increasing people's confidence and trust in organisations [28, 73, 74].

The notion of information being visible or open to stakeholders is useful in software engineering. It is related to one aspect of communication activities in the software life cycle, which is important to the success of software projects. The main goal of transparency is to make information visible to stakeholders so that they can evaluate a software system or to make decisions based on the visible information.

In the simple communication model, transparency implies that the receiver should be able to answer his or her questions using the information from the channel. The receiver encounters transparency problems when he or she cannot understand the information that he or she needs from the communication channel. However, how the receiver understands information is ambiguous. It is also unclear how the sender or the channel should make information transparent to the receiver.

In this thesis, we explore notions of transparency and evaluate its usefulness in the context of software engineering. The following section presents our research questions and research approach.

1.6 Research Approach

The goal of the thesis is to conceptualise transparency in the software engineering context. We aim to collect evidence to answer three questions about transparency and its usefulness in software engineering. We believe that accessibility, understandability, and relevance are important for achieving transparency in software development. We also believe that transparency is a useful concept that helps stakeholders to see the information necessary to achieve their goals. If the sender of information communicates to the receiver with the concept of transparency in mind, the information communicated to the receiver can be improved. The receiver can access and understand the information needed to answer his or her questions. A formal definition of transparency will enable a diagnostic framework based on the definition. This framework will help developers to articulate problems with communicating information in software systems and projects.

This thesis aims to answer the following research questions:

- RQ1. How much does the term "transparency" occur in the software engineering literature?
- RQ2. What is the concept of transparency in the software engineering context?
- RQ3. How important is the concept of transparency to successful software development?

To answer these research questions, we divide the research into the following stages (Chapter 4 discusses our research approach in detail):

1. Exploration.

   The exploration stage of the research addresses research questions, RQ1 and RQ2. We conducted a literature review of the definitions of transparency from different fields. We also looked at how transparency was defined in different aspects of software engineering. In addition, we conducted an exploratory survey which collected opinions about communication problems and definitions of transparency from software practitioners.

2. Evaluation.

   The evaluation stage of the research aims to answer RQ3. We evaluated our evidence about the usefulness of the concept of transparency in software engineering. This evaluation helps us to gain confidence about the importance of transparency in software engineering.

   To answer RQ3, a set of hypotheses was derived from the literature review and the survey findings. The set of hypotheses helped us to identify the scope for the evaluation. In this thesis, we began to answer RQ3 by testing one of the hypotheses. We conducted an experiment for comparing the effectiveness of two different requirements documents with different degrees of transparency in presenting functional requirements of a software system. We wanted to see if a more transparent software artefact would be more effective in enabling stakeholders to answer questions about a software system than a less transparent software artefact.

3. Application.

   The application stage of the research aims to apply the concept of transparency in software engineering practice. This stage is the future work of our research. The long term goal is to formalise the concept of transparency in software engineering and to introduce a diagnostic framework based on our definition of transparency for developers to improve communication in the software life cycle.

1.7 Thesis Overview

This thesis is organised as follows:

In Chapter 2, we present more detailed definitions of transparency in different areas. We describe notions of transparency from philosophy, organisations, business ethics, public participation, and computing orientations. In this chapter we understand how transparency is used in different areas and identify what concepts are related to transparency before we define transparency in software engineering. In Chapter 2, we also discuss implications of transparency useful to software engineering.

In Chapter 3, we propose a working definition for transparency in software engineering. We discuss the three attributes of transparency: accessibility, understandability, and relevance. All of them are important for achieving transparency in software engineering. Moreover, we discuss the assumptions underpinning the working definition. The definition is tentative ('working') because we are exploring the notions of transparency in software engineering. The working definition helps us interpret transparency's usefulness for improving communication in the software life cycle. Chapter 3 also has an overview of how transparency is used in software engineering.

In Chapter 4, we describe our research approach in more detail through three main stages: exploration, evaluation, and application. In this thesis, we focus on the exploration and the evaluation stage. The application stage will be the future work, which will involve the introduction of a diagnostic framework based on transparency to software developers. In this chapter we also identify the scope for evaluating the usefulness of transparency in software engineering.

In Chapter 5, we present our exploratory survey for collecting personal opinions from software practitioners about communication problems and transparency in software engineering. This survey enables us to gain insights into different communication problems in software projects and to improve our definition of transparency for software engineering. The results of the survey and the threats to its validity are also presented in this chapter.

In Chapter 6, we present an experiment for evaluating the importance of transparency in requirements engineering. In this experiment we study the effectiveness of two different requirements documents with different degrees of transparency in presenting functional requirements of a software system. The evidence collected from the experiment helps us to support or refute our hypotheses about the usefulness of transparency to software development. In this chapter, we also discuss the results and validity threats to the experiment.

Chapter 7 has discussion of our research findings. We revisit our research questions and discuss interesting points from our findings. This chapter also explores limitations of the research and discusses improvements for the survey and experiment.

Chapter 8 is a summary of the thesis and contributions of our research. In this chapter we summarise our findings from each chapter of the thesis. We also summarise the contributions of our research. Finally, we suggest areas for future research and conclude with thoughts about transparency in software engineering.

2
Definitions of Transparency

This chapter presents an overview of transparency's usage in philosophy, organisations, business ethics, public participation, and computing. This helps us to gain insights into transparency's applications in the software life cycle. We also discuss implications of transparency from these areas in terms of the simple communication model. Notions of transparency from these areas are associated with the perception of people, the disclosure of information, or the hiding of information rather than the physical property of an object. We limit our discussion to philosophy, organisations, business ethics, public participation, and computing in relation to transparency.

2.1 Transparency in Philosophy

The areas of philosophy of the mind, epistemology, and philosophy of language use the term "transparency". We discuss transparency in these philosophical areas.

In the philosophy of the mind, transparency describes the phenomenon of an individual's perceptual experience, known as the "transparency of experience". The transparency of experience is related to the process of introspection where

> "one apparently looks through the experience to the world, just as if the experience itself were transparent" [107].

In epistemology, transparency is related to an individual's self-knowledge, which is the knowledge of an individual's mental states [47]. The mental states include an individual's beliefs, desires as well as sensations [47]. According to Boghossian [12], transparency plays an important role in epistemological arguments for the study of self-knowledge. Transparency describes the apparent privileged access of mental states where an individual knows about facts or features of those states [107].

In the philosophy of language, transparency is known as "referential transparency", which is the opposite of "referential opacity". Referential transparency means that

> "the truth about a given object is not usually affected by the manner of referring to it" [53].

This concept of transparency is also applied in computer science for understanding programming languages. Referential transparency refers to the property of a function in which the return value of the function is always the same regardless of where the evaluation occurs [31].

Although transparency is an important concept in philosophy, it is unclear how the implications of transparency help to improve communication in the software life cycle. Transparency in philosophy is related to a receiver of information who observes the world; transparency depends on the experience of knowledge of the receiver. However, in the context of our communication model, it is unclear who the sender is, what communication channel is used, or what "information" the receiver needs.

In summary, transparency in philosophy has different notions depending on the context. It can relate to an individual's perception or an individual's mental states. Transparency in philosophy can also relate to how an individual refers to an object.

2.2 Transparency in Organisations

The public sphere stresses the importance of transparency for open information and operations in governments and organisations. For example, Transparency International, a non-governmental organisation that monitors corruption in governments and large corporations, promotes transparency to fight corruption in international transactions. Transparency International defines transparency as

> "a principle that allows those affected by administrative decisions, business transactions or charitable work to know not only the basic facts and figures but also the mechanisms and processes. It is the duty of civil servants, managers and trustees to act visibly, predictably and understandably" [112].

This definition has several implications. It suggests who the receiver of information is and what information to disclose to the receiver. The receiver's questions are affected by governments or organisations. The sender (civil servants, managers, trustees, etc.) is responsible for providing information to the receiver. The receiver should access information easily and find it useful.

In another definition of transparency, Oliver [84] argues that transparency is necessary for success in today's society. Transparency influences the reputation and the credibility of large organisations and business practices. This includes governments. Transparency also affects the trust of people in organisations. To prevent organisations from losses and erosion of trust, essential facts about organisations must be open and available to everyone. Oliver observes that transparency consists of three components; an observer, something available for observation, and a method for the observation. He defines transparency as

> "letting the truth be available for others to see if they so choose, or perhaps think to look, or have the time, means, and skills to look."

He argues that transparency has the implication of passive disclosure, in which the truth is available only upon the requests or motives of people. Oliver further suggests incorporating the idea of active disclosure to transparency in organisations. This means that organisations need to provide timely and accurate information to stakeholders as well as getting feedback from stakeholders.

Oliver's definition of transparency implies that receivers should easily access information. The definition also suggests that the transparency of information depends on the time, the means and the skills of the receiver to look for information. In addition, this definition shows the limitation of transparency where the sender's information might not be timely or accurate for the receiver.

Likewise, Lord [67] points out that the information disclosed to the public might not represent the truth about organisations. Transparency is a condition that makes information available to the global public. In the context of international regimes, transparency is about "the availability and accessibility of knowledge and information" [19]. Mechanisms such as open government hearings and mobile phones can enhance transparency. However, the notion of transparency does not guarantee proper disclosure of information by organisations. Oranisations can distort information or reveal information in a way that benefits certain parties [67].

In the area of economics and business, transparency concerns the idea of providing timely and accurate information to the receiver. The Handbook of International Financial Terms [78] defines transparency as

> "a condition of the markets as to the availability and timely dissemination of price and other relevant information to market participants."

This definition of transparency shows, again, that the information should be available and timely to the receiver (market participants). Relevant information about the markets should also be available to the receiver.

Transparency also has implications as a feature of organised exchanges in financial markets. Transparency ensures that participants have accurate information on market prices [78]. In addition, the banking industry uses transparency to improve the disclosure of a bank's activities to the public. Transparency is about the

> "public disclosure of reliable and timely information that enables users of that information to make an accurate assessment of a bank's financial condition and performance, its business activities, and the risks related to those activities" [9].

This definition suggests that the information provided by the sender (banks) should be useful to the receiver (the public) so that the receiver can assess a bank's condition. To achieve a satisfactory level of bank transparency, there are five qualitative characteristics for transparent information. These are comprehensiveness, relevance and timeliness, reliability, comparability, and materiality [9].

In summary, transparency in organisations is mainly concerned with making information about an organisation visible to stakeholders. It is important for organisations to make information easily available and accessible to stakeholders. However, the definitions for transparency in organisations reveal that not all information is properly disclosed by organisations. Organisations also need to provide useful information to stakeholders; useful information includes timeliness, accuracy, and reliability to enable stakeholders to assess the organisations.

2.3 Transparency in Business Ethics

Transparency in the literature of business ethics is found to be an important value to organisations. For example, Dubbink et al. [34] discuss transparency as an important condition for corporate social responsibility. In a special issue of Ethics and Information Technology [116], the concept of transparency is presented by scholars and practitioners for business ethics. The special issue also presents a discussion by Turilli and Floridi [115] which defines transparency in the context of computer ethics: "the possibility of accessing information, intentions or behaviours that have been intentionally revealed through a process of disclosure".

Turilli and Floridi further discuss "information transparency" as a pro-ethical condition for enabling or hindering ethical principles such as accountability, privacy and copyright. Information transparency concerns enabling stakeholders to make informed decisions. It is also about enabling organisations to demonstrate to their stakeholders that the organisations are complying with legal requirements and ethical principles. To achieve information transparency, both the information and the information's production must be disclosed. Moreover, Turilli and Floridi assert that the information disclosed should have "meaningful, veridical, comprehensible, accessible and useful data".

In business ethics, transparency means "corporate transparency" which is related to the disclosure of information through standardised reporting. The disclosure of information is generally unidirectional, from an organisation to its stakeholders [117]. To improve corporate transparency, Vaccaro and Madsen [117] introduce the concept of "corporate dynamic transparency" which implies a two-way communication between an organisation and its stakeholders. According to Vaccaro and Madsen, organisations should provide customised information and internet-based tools that enable organisations and their stakeholders to exchange information. Similarly, Cohen and Hiller [21] also discuss a two-way collaborative model for corporate transparency.

In addition, Elia [38] discusses corporate transparency in relation to stakeholders' rights. Stakeholders' rights concern the needs and ethical expectations of stakeholders. Elia argues corporate transparency should aim to protect stakeholders' interests as well as to enforce corporate social responsibility. Elia suggests that organisations should follow the theory of stakeholders' rights when disclosing information to their stakeholders.

In the special issue of Ethics and Information Technology [116], Menéndez-Viso [72] presents an overview of notions of transparency from Western philosophy and literature. The first definition of transparency presented has the meaning of invisibility where an individual's actions go unnoticed to others. Transparency was desirable to offenders in ancient societies as others could not see the malicious acts conducted by offenders. However, transparency nowadays refers to the notion of enabling people to see the information about organisations.

Menéndez-Viso further argues that the current notions of transparency are insufficient to achieve corporate transparency. This is because the modern claims for transparency are claims for appearance where agents providing the information are not invisible but apparent. The environment in which agents act is transparent (i.e. invisible) so that the information is not hidden. Furthermore, Menéndez-Viso discusses that transparency entails more than the disclosure of information, it also involves the production of information, in which the information produced should be good and useful to people. However, the quality of information depends on the producer of information. For example, documents created by an organisation about its activities might not reflect a true or fair image of the organisation. Hence, honesty, integrity and public care are necessary to enable stakeholders to evaluate an organisation's activities. Menéndez-Viso suggests that the opacity of information should be considered when demanding transparency in organisations. The question for transparency as discussed by Menéndez-Viso is, therefore, "what do we need and want to see, and how is this going to be produced?"

The definitions of transparency in computer ethics and business ethics imply that the information should be readily available and easily accessible to the receiver of information. Moreover, information should be useful to the receiver. Menéndez-Viso's question for transparency implies that the sender's information should be related to what the receiver needs and wants. Although the notion of transparency is important in business ethics, transparency is limited in providing information that faithfully represents the truth about an organisation. This limitation is similar to the one discussed previously about transparency in organisations. The sender (organisations) has the ability to control what information the receiver sees. The information produced by the sender might not reflect what an organisation is actually doing or how well it is performing.

2.4 Transparency in Public Participation

Public participation is an important practice for government projects, particularly projects that concern large infrastructure development. It consists of activities for informing the public about government projects. It also concerns feedback on proposed plans [10, 91].

As one criterion for effective public participation, Rowe and Frewer [91] say

> "the participation process should be transparent, so that the wider public can see what is going on and how decisions are being made".

Rowe and Frewer also use transparency in a framework to evaluate public acceptance of different public participation methods. Rowe and Frewer suggest that transparency could be the principle for revealing the process of selecting participants. Transparency should also inform the public about final decisions made in government projects. Furthermore, Rowe and Frewer propose that transparency should be one of the criteria for decision-making. Better transparency occurs by documenting the decision-making process and the outcomes of the decisions made.

Similarly, Bickerstaff et al. [10] propose transparency as one of four key principles of public participation in transport planning. Transparency is used to identify the influence of public involvement on local transport planning. For them, transparency is

> "the degree to which the outputs and impacts arising as a result of participation are explicitly reported, demonstrated and fed back to the participants" [10].

In summary, transparency in public participation is useful for evaluating the effectiveness of public participation methods. Transparency is also useful for providing the public an indication of how much influence the public has on government planning. Similar to the notions of transparency in business ethics, transparency in public participation implies that the information is accessible to the receiver (the public). The sender (governments) provides information that is related to the process of participation and the outcome of the participation.

2.5 Transparency in Computing

The term "transparency" appears in various subfields of computing. Transparency has different meanings depending on the context. We find 10 related entries in the subject of computing through a search for the term "transparency" in entry headings in the Oxford Online Reference. Example entries include network transparency from the Dictionary of Computing [31] and referential transparency from the Dictionary of the Internet [54].

Transparency can be used in computer graphics as a technique for making objects translucent [31]. Transparency also concerns a property of a network or a distributed system, with the notion of being invisible to users in a network. For example, transparency is defined as "the property that makes the user of a network unaware of the fact that they are interacting with a network" in the Oxford Dictionary of the Internet [54]. Similarly, transparency in a distributed system is about users' inability to distinguish whether they are accessing local resources or remote resources [108]. In addition, Farooqui et al. [40] provide a summary of transparency mechanisms of the ISO reference model for open distributed processing. Here, transparency means "distribution transparency" which has the implication of hiding objects, mechanisms, or boundaries from clients or users. For example, location transparency, as one of distribution transparencies, "hides from a user (client) where the object (server) being accessed is located" [40].

In other areas of computing, transparency has similar notions to those of organisations, business ethics and public participation. For example, transparency in risk management is "a condition that all functions of software are disclosed to users" [74]. Transparency in this example has the connotation of visibility and openness. The remainder of this thesis is devoted to explore transparency with this connotation in the context of software engineering. We will discuss how transparency with this connotation is used in software engineering in the next chapter.

2.6 Discussion

From the review of the literature, we identify three themes that are useful to communication in software engineering. The first theme of transparency involves the notion of making information available and accessible to the receiver. This notion is important for communication during the software life cycle. The receiver can evaluate software systems using the information obtained in the software development process. The receiver can also make decisions based on such information.

Secondly, the theme of providing comprehensible or understandable information to the receiver is important to software development, particularly in the process of requirements elicitation and negotiation. Understandable information enables the receiver to assess whether software systems are meeting his or her expectation.

The third theme is concerned with the notion of providing relevant information to the receiver's needs to improve communication in the software life cycle. Communication problems such as information overload can be minimised. The time spent for the receiver in finding relevant information can also be reduced.

In addition, we identify some restrictions for transparency from the literature. Transparency depends on the sender who controls what information the receiver sees. The information provided by the sender might not be timely or accurate. The information might also be distorted in a way that benefits certain parties at the expense of informing receivers. Moreover, transparency is meaningful only when the receiver communicates with the sender. The notions of transparency suggest that the time, the means, as well as the receiver's skill on accessing information all affect how well the receiver communicates with the sender.

2.7 Summary

In this chapter, we present definitions of transparency as they relate to different areas. Notions of transparency vary depending on the context in which transparency is used. In philosophy, transparency is related to our perception or mental states. Transparency is also related to how an individual refers to an object. In the context of organisations, business ethics, and public participation, transparency plays important roles. It has the meaning of information disclosure. Paradoxically, transparency has the notion of invisibility when used in the context of network or distributed systems.

From the literature, we identify three themes of transparency that are useful to communication in software engineering:

- the notion of making information available and accessible to the receiver,
- the notion of providing comprehensible or understandable information to the receiver, and
- the notion of providing relevant information to the receiver's needs.

We also identify some restrictions of the concept of transparency. For example, transparency depends on the sender who controls the information communicated to the receiver.

In the next chapter, we present a working definition of transparency in software engineering. We also review how transparency is used in areas that are related to software engineering.

3

Transparency in Software Engineering

In this chapter we first present a working definition of transparency for software engineering. The definition is under development as we explore transparency and its functions in software engineering. The working definition helps us to observe and to describe how transparency affects communication in the software life cycle. The long-term goal of the research is to establish and formalise the definition of transparency with evidence from multiple sources such as literature reviews and empirical studies. This chapter also presents an overview of the existing notions of transparency in software engineering. We discuss how our definition fits within the existing notions of transparency in software engineering.

3.1 A Working Definition for Transparency

Many notions of transparency as we discovered in Chapter 2 concern how well receivers see and use the information from senders. In the context of software engineering, transparency should help stakeholders to see and use the information communicated to them during the software life cycle. How well stakeholders see and use the information depends on the degree of transparency in the communication channel. Therefore, we define transparency in software engineering as:

> *the degree to which stakeholders can answer their questions by using the information they obtain about a software system during its life cycle.*

In this definition, stakeholders refer to anyone involved in the development of a software system. Example stakeholders are software developers, project managers, clients, and end users.

Stakeholders need three attributes of transparency to find answers to their questions. These attributes are based on the implications of transparency discussed in Chapter 2. The first attribute, accessibility, concerns the ability of stakeholders in obtaining information from a sender of information. Information held by the sender may range from one bit to many sets of data that contain millions of bits in different communication channels. Once stakeholders obtain any data, they can assess whether such information answers their questions. To decide if the information answers their questions, stakeholders must first understand the meaning of the information. This is our second attribute of transparency, understandability. The third attribute is relevance, which is concerned with how well stakeholders can answer their questions using the information. Transparency's usefulness involves accessibility, understandability, and relevance. These three attributes are important for enabling stakeholders to see the information that they need to answer their questions.

In the following subsections, we discuss the three attributes of transparency: accessibility, understandability, and relevance. We also discuss the assumptions made for our working definition.

3.1.1 Accessibility

The term "accessibility" appears in many software engineering-related areas such as requirements engineering [13, 80], HCI [23, 92, 123], and on-line information services [26]. Accessibility is often associated with the usability of an application, where anyone can access or use such application. For example, in HCI accessibility is "having access to the products needed to accomplish a goal" [92].

Accessibility also concerns how well users can retrieve information. According to Zaki and Forbrig [123], accessibility is

> "the opportunity for all the users to receive and to deliver all kinds of information, regardless of the information format or the type of user impairment."

Similarly, accessibility, according to Culnan [26], is related to

> "...the ability to retrieve the desired information successfully..."

For this thesis transparency must concern how well stakeholders can access information to answer their questions. Therefore, we say accessibility is

> *the degree to which stakeholders can obtain information that they believe is likely to answer their questions easily.*

To determine how well stakeholders access such information, we need to answer three questions.

1. Is the communication channel available for stakeholders to find answers to their questions?

   In our communication model, the sender provides communication channels where stakeholders find answers to their questions. The communication channels can be documents, pictures etc. The sender can also be the channel that stakeholders communicate with directly. In order for stakeholders to find answers to their questions, the communication channel should be available to them.

2. How easily can stakeholders use the format in which the information is presented?

   The format of the information should be easily usable by stakeholders so that they can find answers within the channel. The format of information depends on the type of the channel. For example, the channel may be an electronic file formatted to open only by Microsoft Word.

3. How easily can stakeholders access information from the channel that they believe is likely to answer their questions within a reasonable amount of time?

   The channel may contain one bit of information or many sets of data. Stakeholders should be able to obtain their desired bit of information quickly. If the channel contains many sets of data, stakeholders should be able to reach the right location of a particular set of data within a reasonable amount of time.

Accessibility is the primary attribute of transparency because stakeholders need to first obtain some information from a channel before assessing whether such information is helpful. After stakeholders have access to the desired information, understandability and relevance of such information can be assessed.

3.1.2 Understandability

Understandability is the second attribute of transparency for assessing the information obtained by stakeholders. When stakeholders access the desired information, they should be able to understand the information before using it. Understandability or comprehensibility in software engineering often concerns the quality of software artefacts. It is also a factor that affects the usability of a software product [48]. Research on understandability is in studies such as usability of computer documentation [50], requirements engineering [22, 109], conceptual modelling [6, 85], and program comprehension [14]. An example definition of understandability in requirements engineering is:

> "the degree to which information contained in a SRS [software requirement specifications] can be easily understood by a reader of that SRS" [22].

In this thesis, we are interested in how well stakeholders understand the information presented in the channel. Therefore, we say understandability is

> *the degree to which the information obtained by stakeholders can be comprehended with prior knowledge.*

To determine how well stakeholders understand the information presented in the channel, we need to ask the following question:

- Once stakeholders obtain the information, how easily can they recognise the meaning of the information within a reasonable amount of time?

  The one-bit of information or a set of data should be understandable to stakeholders so that they can assess if this information answers their questions. The time needed for stakeholders to understand the information depends on their prior knowledge.

Understandability is important to transparency because, without it, stakeholders cannot assess whether the information answers their questions. If stakeholders understand the information, the next issue is how well the information answers stakeholders' questions. This leads to the third attribute of transparency, relevance, which follows in the next section.

3.1.3 Relevance

Relevance is the third attribute of transparency for assessing stakeholders' ability to find answers to their questions from the received information. The term "relevance" is often used in information search and retrieval and refers to

> "an evaluation of the match between a question (or search statement) and the answer (or text) retrieved by that statement" [16].

According to Case [16], relevance in information science concerns the technical measures of document retrieval. Relevance depends on the question or the search statement which can be measured by precision and recall. In areas such as psychology, relevance involves the context in which an individual is situated. Relevance depends on an individual's knowledge state and his or her intentions at the time he or she encounters the information [16].

Relevance in software engineering refers to the technical measures of information retrieval or individuals' subjective judgements of the information. Research on relevance in software engineering can be found in studies and discussions such as cost estimation for software projects [58], source code search engines [70], and software documentation [3, 69].

In this thesis, we want to know how well stakeholders can answer their questions using the information from the channel. Therefore, we define relevance as

> *the degree to which the information obtained by stakeholders answers their questions.*

To determine how well the information from the channel answers stakeholders' questions, we ask the following:

- How well can stakeholders use the obtained information to answer their questions within a reasonable amount of time?

Judging the relevance of information depends on stakeholders who obtain the information. This consideration leads to the following questions.

- How quickly can stakeholders answer their questions using the information?

  This question concerns the time needed for stakeholders to use one bit of information or a set of data to answer their questions. The time it takes for stakeholders to answer their questions also depends on their prior knowledge.

- How directly connected is the information with stakeholders' questions?

  We are interested in whether the information is directly connected to stakeholders' questions. The opposite of "directly connected" is circumlocution. The use of circumlocution might affect the understandability of information. It might also increase the time for stakeholders to assess the information to answer their questions.

- Does the information answer stakeholders' questions sufficiently?

  The last question addresses stakeholders' need to find the current channel suitable or to seek another channel or location within the channel to answer their questions.

The assessment of the information also depends on stakeholders' prior knowledge and expectations for the information.

Relevance is an important attribute of transparency so stakeholders can answer their questions within a reasonable amount of time only if the information is relevant to their questions. However, stakeholders can assess the relevance of information only after they access and understand the information. The degree of transparency thus depends on the accessibility, understandability, and relevance of information. Because we are developing the concept of transparency in software engineering, our definition is necessarily restricted. In the following section, we discuss the assumptions made for our working definition of transparency.

3.1.4 Assumptions

We argue that transparency enables stakeholders to answer their questions about a software system using the information they obtain in the software life cycle. Transparency's components are accessibility, understandability, and relevance. However, our working definition is restricted as it does not specify the behaviour of the sender of information or the truthfulness of information provided to stakeholders. Our definition depends on the judgement of stakeholders who obtain the information during the software life cycle. We base our definition on the following assumptions for transparency:

- Information held by the sender is not falsified or distorted in a way that benefits certain parties. We assume that the sender communicates with receivers (stakeholders) in good faith. The sender has no malicious intentions.

- Receivers have the need to know about a software system during its life cycle. To satisfy their need to know, they have a set of questions in their mind. The type of questions depends on the context in which receivers are situated. Receivers evaluate the degree of transparency when they make query about the software system. Transparency has no meaning if receivers do not make query about the system.

- Receivers have reasonable expectations for information about the software system. They also have reasonable expectations for the time that they spend on obtaining and assessing information to answer their questions. Receivers' expectations can be affected by different factors such as their background and environment. We assume that the receivers are reasonable stakeholders. They have legitimate questions about the software system during its life cycle. Moreover, they are competent in their roles and are not extreme stakeholders who are never satisfied with the sender or the information that they receive.

In the following section, we give an overview of how transparency is used in software engineering. We also discuss how accessibility, understandability, and relevance are related to the existing notions of transparency in software engineering.

3.2 Transparency in Software Engineering

An overview of the existing notions of transparency in software engineering shows a breadth in software engineering-related contexts. We focus on areas in software engineering where the notion of transparency concerns visibility or openness of information. In each subsection, we further discuss how our definition and the three attributes (accessibility, understandability, and relevance) are related to existing notions of transparency.

3.2.1 Transparency in Information Privacy

Information privacy is concerned with an individual's control over his or her personal information that is held by third parties. Transparency provides a means of privacy protection for individuals by allowing them to monitor their data as well as actions of others [20]. In the context of information privacy, transparency refers to the accessibility of information about individuals' personal data and the usages of such data. For example, transparency is defined by Awad and Krishnan [8] as the ability of consumers having "access to the information a firm has collected about them, and how the information is going to be used".

Transparency is also a means for facilitating information accountability, which is another mechanism for protecting information privacy. Transparency enables individuals to see the use of their personal information so that people and organisations can be held accountable for any misuse of information [120]. The following quote as an example definition of transparency in relation with privacy and accountability for E-health systems:

> "information held about the consumer is visible to the consumer ... and so is the use (access to) of that information by anyone else so that any action could be tracked back to an individual" [46].

In the context of information privacy and accountability, the existing notions of transparency involve the accessibility of information. To be transparent, the sender of information needs to make information accessible to stakeholders. This information should answer stakeholders' questions about all potential usages of personal information held by third parties.

3.2.2 Transparency in Computer Ethics

Transparency in computer ethics is a means to assure validity of information [96] and a means to support stakeholders in the decision-making process [42, 43]. It is also an important ethical value to the process of computational modelling [43]. According to Turilli and Floridi [115], transparency in information management, business ethics and information ethics is

> "the possibility of accessing information, intentions or behaviours that have been intentionally revealed through a process of disclosure".

Turilli and Floridi frame transparency as "the choice of which information is to be made accessible to some agents by an information provider". Turilli and Floridi further discuss the ethical nature of information transparency as it relates to the disclosure of information. Transparency depends on factors such as availability, accessibility of information, as well as ways that information supports decision-making. Moreover, transparency depends on decisions of information providers about what information to disclose and appropriate forms to disclose such information. Turilli and Floridi explain that the information disclosed should consist of "meaningful, veridical, comprehensible, accessible and useful data". In addition, Turilli and Floridi argue that details of the information's production should also be disclosed to enable ethical principles that affect or regulate the disclosure of information.

The term "comprehensible", used by Turilli and Floridi to describe the characteristics of disclosed information, suggests that understandability of information relates to their definition of transparency. Similarly, Fleischmann and Wallace [43] describe transparency as enabling stakeholders to use computational models to evaluate information and then make informed decisions. Fleischmann and Wallace's definition of transparency is

> "the capacity of a model to be clearly understood by all stakeholders, especially users of the model."

The literature shows that existing notions of transparency in computer ethics are concerned with the accessibility and understandability of information. Transparency is an ethical value which is important in computer ethics as the information provided by the sender helps stakeholders to make informed decisions. The type of information that stakeholders need depends on the answers to their questions to make decisions.

3.2.3 Transparency in Security, Trust and Risk Management

Transparency has the notion of visibility or information openness when it appears in discussions on security and trust of software systems. Transparency provides a means of security assurances to stakeholders about software systems through the openness of information [73]. The degree of transparency of a software system affects how much stakeholders trust the system [28, 73]. Similarly, transparency in risk management refers to the disclosure of information to stakeholders. It is a means to proper risk management. Transparency is "a condition that all functions of software are disclosed to users" [74].

Transparency is also a means to increase trust and acceptance of user-adaptive systems [25]. Transparency helps stakeholders, specifically end users, to understand how a system works. According to Cramer et al. [25], a transparent system "allows the user to understand the way it works and explains system choices and behaviour".

Existing notions of transparency in the literature suggest that transparency influences security assurances, trust, and risk management of software systems. Notions of transparency in these areas concern accessibility and understandability of information. Stakeholders with questions about a software system should find information from the sender accessible and understandable, which makes the information usable.

3.2.4 Transparency in Visual Notations

Transparency refers to the understandability of information used in the context of visual notations or graphical representations. It is one of principles for designing cognitively effective visual notations. Moody [79] calls transparency "semantic transparency" which has "the meaning of a symbol [which] can be inferred from its appearance". This definition of transparency implies that stakeholders can easily recognise the meaning of visual notations. It also suggests that stakeholders are able to understand the information presented in diagrams that consist of semantically transparent visual notations.

In the context of our simple communication model, the visual notations are the channel for communicating information to stakeholders. Stakeholders might have questions about what the visual notations represent or what information the visual notations convey about a software system. Although accessibility and relevance are important attributes to our definition of transparency, they are not apparent from the existing definition. It appears that the understandability of information is the most important attribute to transparency for visual notations.

3.2.5 Transparency in Agile Development

Transparency is often a value or principle for agile development. Transparency can be achieved through agile practices such as stand up meetings and arrangements of work areas that maximise easy information sharing [55]. According to Bird [11], one principle of agile methods is to maintain honesty and transparency, where it is difficult to hide problems in software projects. Software developers share information about the software project with customers so that they can make decisions early to mitigate any problems.

Similarly, transparency is a supporting pillar for process control in Scrum, an agile development framework [97]. Schwaber and Sutherland explain that

> "significant aspects of the process must be visible to those responsible for the outcome. ... Transparency requires those aspects be defined by a common standard so observers share a common understanding of what is being seen."

These discussions on transparency suggest that transparency in agile development is concerned with the accessibility of information. The notions of information sharing and making aspects of a process visible have the implication that stakeholders have access to information. Moreover, the existing notions of transparency in agile development are concerned with the understandability of information. The notion of using a common standard for common understanding implies that stakeholders can easily interpret the meaning of information visible to them. The type of stakeholders' questions might be related to the development process of a software system and the management of a software project.

3.2.6 Transparency in Dependable Systems

Transparency is one proposed approach for developing dependable software systems. It is a means for enabling stakeholders to assess the dependability of a software system [56]. Jackson et al. [56] explain that

> "customers and users can make informed judgments when choosing suppliers and products only if the claims, criteria, and evidence for dependability are transparent."

Enabling stakeholders to make informed judgements is similar to the notions of transparency in computer ethics discussed previously. Transparency in dependable systems is concerned with the understandability and relevance of information. Information with such claims for dependability should be understandable so that stakeholders can judge the relevance of information. Information should also be relevant so that stakeholders can base their decisions about the choice of suppliers and products on the information.

3.2.7 Transparency in Requirements Engineering

In requirements engineering, transparency is a general quality or a non-functional requirement related to the disclosure of information [94, 95, 104]. In addition, transparency spreads to different parts of a software system [95]. Transparency, as discussed by Software Transparency Team [104], enables stakeholders to have

> "accessibility, usability, informativeness, understandability and auditability of information [or processes] held by centers of authority (society or organizations)".

Sampaio do Prado Leite and Cappelli [95] stress the importance of transparency for software systems. They explore notions of transparency and represent transparency using a non-functional requirements framework. They also use a Softgoal Interdependence Graph (SIG) to illustrate correlation and contribution links between transparency and different quality attributes. Sampaio do Prado Leite and Cappelli identify 33 quality attributes that contribute to transparency. The 33 quality attributes are based on information sources that Sampaio do Prado Leite and Cappelli have found. They have selected 3 sites, 5 books, and 1 scientific paper as information sources. The five main quality attributes are accessibility, usability, informativeness, understandability, and auditability in the transparency SIG. Each main quality attribute also composes other quality attributes. For example, portability, availability, and publicity are quality attributes that contribute to accessibility.

Serrano and Sampaio do Prado Leite [98] further investigate the quality attributes of transparency proposed by Sampaio do Prado Leite and Cappelli [95]. Serrano and Sampaio do Prado Leite refer to the quality attributes as requirements patterns and capture arguments from stakeholders regarding applications of the concept of transparency for a given software system. From the 33 quality attributes of transparency, Serrano and Sampaio do Prado Leite emphasise the importance of accessibility to transparency because other quality attributes cannot be achieved without access to a software system.

It seems that our three attributes of transparency (accessibility, understandability, and relevance) are among the 33 quality attributes. However, when examining the list of the quality terms and the definitions provided by Sampaio do Prado Leite and Cappelli [95], we find that our definition of relevance is not one of the 33 quality attributes for helping transparency. Our definition for accessibility seems to relate to the definitions of accessibility and informativeness provided by Sampaio do Prado Leite and Cappelli. Accessibility, according to Sampaio do Prado Leite and Cappelli, is "the quality of being easy to meet deal with", and informativeness is "the quality of providing or conveying information". These two definitions are relevant to our definition of accessibility as they concern how easy it is for stakeholders to obtain information, and how well the information is conveyed to the stakeholders. Our definition for understandability is also similar to their definition of understandability, which has the meaning of "the quality of comprehensible language or thought".

Although two of our attributes of transparency are among the 33 quality attributes, we find that their definitions for the quality attributes are ambiguous and the context for the definitions is unclear. An example of this is the definition for operability, which is one of the attributes that contribute to usability. It has the meaning of "the quality of being treated by surgical operation". It is unclear from the literature how being treated by surgical operation helps the transparency of software systems. Moreover, many of the quality attributes seem to be dependent on the type of questions that stakeholders have. For example, one of the five main quality attributes, auditability, has the meaning of "the ability to examine carefully for accuracy with the intent of verification". If stakeholders do not intend to verify information, then auditability is not relevant to stakeholders' questions and thus auditability will not affect the degree of transparency.

Transparency in the context of requirements engineering is a quality or a non-functional requirement for a software system that helps to disclose information to stakeholders. The existing notions of transparency in requirements engineering are linked with many different quality attributes or non-functional requirements. Accessibility and understandability are among the 33 quality attributes discussed by Sampaio do Prado Leite and Cappelli [95]. However, it is unclear how their quality attributes affect how well the information is disclosed to stakeholders. The type of stakeholders' questions is also ambiguous in the literature.

3.2.8 Transparency in Other Software Engineering Areas

Transparency can be found in other software engineering-related areas. For example, transparency or visibility is a software quality which can be an internal or external quality [48]. According to Ghezzi et al. [48], transparency makes a development process available and easily accessible for examination. Transparency benefits software development as it helps developers to assess the impact of their actions and to make decisions. Ghezzi et al. further discuss that a product is visible if

> "it is clearly structured as a collection of modules, with clearly understandable functions and available and accurate documentation".

Transparency also supports communication and coordination behaviours in software development as it makes work visible to stakeholders [30]. According to Dabbish et al. [30], transparent development environments allow everyone to "see and have meaningful access to (almost) everything".

The existing notions of transparency involve visibility to stakeholders which suggests that the accessibility of information is necessary for stakeholders to use information. The sender of information needs to provide information that answers stakeholders' questions about a development process or any work done during software development. In addition, understandability of information is also important for transparency in software development as developers need to examine a development process using the information visible to them.

3.3 Summary

Our working definition of transparency for software engineering has three attributes of transparency: accessibility, understandability, and relevance. Moreover, we review how transparency is used in different software engineering areas.

From the review of the literature, we find that the notions of transparency are useful to software engineering. Transparency is used as a means to enhance privacy and accountability. It is also used to improve trust in software systems. Furthermore, transparency is a means to help stakeholders to make decisions in software development. Although much software engineering literature promotes ideas of transparency, the term is not well defined or explicitly assessed in software engineering. It is unclear what constitutes transparency or how transparency can be assessed. It is also ambiguous about transparency's use in software engineering.

In the next chapter, we present our research approach for exploring transparency in software engineering. We also discuss our approach to evaluate the usefulness of transparency in software engineering.

4
Research Approach

In this thesis we argue that transparency is a useful concept for improving communication in software development. The long-term goals of the research are to formalise transparency and to build a diagnostic framework for developers to articulate communication problems in the software lifecycle. Our research into transparency has three stages:

1. Exploration.

   The first stage of the research focuses on exploring what transparency is and how it is used in different areas. This stage is fundamental for defining transparency in software engineering. It helps us to define a clear picture of transparency and its boundaries in software engineering. In Chapter 2 and Chapter 3 we explore different notions of transparency from different areas. In Chapter 5 we explore different communication problems encountered by software practitioners. In Chapter 5 we also present opinions from software practitioners about our definition of transparency for software engineering.

2. Evaluation.

   The evaluation stage aims to answer the question: how useful is transparency to software development? Answers to this question help us to gain confidence about the value of transparency in software engineering. Findings from the evaluation provide evidence to support or refute our hypotheses about the usefulness of transparency to software development. In this thesis, we evaluate the effectiveness of software artefacts with different degrees of transparency in helping software practitioners and tertiary students to answer questions. The design and the results of the experiment are presented in Chapter 6.

3. Application.

   The application stage aims to apply our definition of transparency in software engineering practice. This stage concerns the future work of our research where we plan to introduce a diagnostic framework on the basis of our definition of transparency to software developers. The objective of this stage is to validate whether the diagnostic framework is useful for developers to articulate communication problems in the software life cycle.

To construct a diagnostic framework for developers to use in software engineering practice, we first collect information about what transparency is by means of empirical methods. Empirical methods can be used for collecting evidence that determines the validity of proposed solutions [36, 60]. Moreover, our theory of transparency can satisfy one of criteria of a good empirically based theory [100, 102] by showing supporting evidence collected from empirical methods.

In this thesis we concentrate on the exploration and evaluation stages of transparency. These two stages are important in helping us to establish a preliminary structure of the diagnostic framework. Findings from these stages are also important to support or refute our definition of transparency. The following sections present our approach to collect information about transparency in the exploration and evaluation stages.

4.1 Exploring Transparency

In the exploration stage we want to gain insights into what transparency is and how it relates to software engineering. These questions guide the research: RQ1 - how much does the term "transparency" occur in the software engineering literature? and RQ2 - what is the concept of transparency in the software engineering context? We do this through literature reviews (Chapter 2 and Chapter 3) and an exploratory survey (Chapter 5).

Chapter 2 and Chapter 3 provide a literature review with an overview of how transparency is defined in different areas such as business ethics and requirements engineering. The literature review helps us to identify attributes that are important to transparency.

The literature review also helps us to understand which definitions of transparency are used in the software engineering literature.

In Chapter 3, we further propose a working definition for transparency in software engineering. We argue that accessibility, understandability, and relevance are the three attributes of transparency that affect how well stakeholders see the information needed to achieve their goals. The working definition enables us to interpret and evaluate transparency in software development.

In addition to the literature review, we design and conduct a survey for collecting personal opinions about transparency in software engineering practice from software project stakeholders. The survey allows us to explore different types of communication problems that stakeholders might encounter in software projects. Findings from the survey help us to improve our proposed definition of transparency and thus answer RQ2. The results of the survey are presented in Chapter 5.

4.2 Evaluating the Importance of Transparency

To answer RQ3 - how important is the concept of transparency to successful software development? - we derive a set of hypotheses from the literature review and the results of the exploratory survey. The set of hypotheses identifies different aspects of software development that could be affected by transparency. It also helps us to identify the scope for the evaluation. The following subsections present the set of hypotheses for RQ3 and the scope of the evaluation.

4.2.1 Hypotheses for RQ3

We derive two main hypotheses from RQ3 which are important to software development. The first hypothesis concerns the development of a software system. The second hypothesis concerns the use of a software system. Each hypothesis is analysed into different hypotheses associated with specific aspects of a software system. We present the two main hypotheses with sub-hypotheses in the following subsection. Because many aspects of a software system such as management, requirements engineering, and testing can be affected by transparency, it is difficult to have a complete set of sub-hypotheses for each main hypothesis. Therefore, we present an overview of the sub-hypotheses for only some aspects of a software system.

Hypothesis A: A transparent development process leads to a successful software project.

(1) A transparent development process leads to a better project management than a non-transparent development process.

  a. Project managers can answer their questions about the progress of a software project in a transparent development process.

  b. Project managers and other stakeholders can answer their questions about the decision-making process as well as the decisions made for a software system in a transparent development process.

  c. Project managers and other stakeholders spend less time having their questions answered in a transparent development process than in a non-transparent development process.

(2) Transparent software artefacts lead to a more effective assessment of a software project than non-transparent software artefacts.

  a. Stakeholders can answer their questions about a software project using the information presented in transparent software artefacts.

  b. Stakeholders spend less time answering their questions using the information presented in transparent software artefacts than using the information presented in non-transparent software artefacts.

(3) Agile development methods are more effective for stakeholders in obtaining information about the project status than non-agile development methods.

  a. Information about the software project is more transparent in stand up meetings than information presented in documentation.

  b. Project managers spend less time obtaining information about the project status in stand-up meetings than using documentation.

(4) Transparent software artefacts lead to a more effective assessment of a software system than non-transparent software artefacts.

  a. A transparent requirements document is more effective for developers in answering questions about the requirements of a software system than a non-transparent requirements document.

  b. A transparent architecture document leads to developers' better understanding of the software architecture than a non-transparent architecture document.

  c. Source code that is transparent helps developers to find bugs during software testing more easily than source code that is not transparent.

Hypothesis B: A transparent software system promotes trust and uptake of the software system by end users.

(5) Transparent software artefacts lead to a more effective assessment of a software system than non-transparent software artefacts.

  a. End users can answer their questions about the functionality of a software system using the information presented in transparent software artefacts.

  b. End users spend less time answering their questions using the information presented in transparent software artefacts than using the information presented in non-transparent software artefacts.

(6) A transparent user interface is easier to use and manage by end users than a non-transparent user interface.

  a. End users can access each function of a software system from a transparent user interface more easily than from a non-transparent user interface.

  b. End users spend less time accessing each function of a software system using a transparent user interface than using a non-transparent user interface.

4.2.2 Scope of the Evaluation

The two hypotheses show our belief in the importance of transparency in different aspects of a software system. To test these hypotheses, we identify different approaches. One of the approaches is to conduct an experiment which helps researchers to determine the validity of proposed theories and methods. Zelkowitz and Wallace [124] discuss 12 experimental approaches from available research methods in software engineering and classify these methods into three categories: observational methods, historical methods, and controlled methods. Similarly, there are three different empirical methods for experimentation in software engineering. The empirical methods are experiments, case studies, and surveys [122].

However, it is not possible to test all hypotheses in this thesis. This is due to the limitations in time, budget, and availability of resources for the research. To test all of the hypotheses, different types of empirical studies would be required. For example, to test hypothesis 3, we would conduct a case study to observe the interactions between project managers and software developers during stand-up meetings. To test hypothesis 6, a controlled experiment could be performed for comparing the usability of a transparent user interface with a non-transparent user interface.

Different types of materials are also needed for different empirical studies. For example, we would need to prepare mock-ups of a software system for evaluating the usability of different user interfaces. We would also need different software artefacts such as requirements documents, source code and user manuals for testing other hypotheses. Furthermore, different groups of participants would be needed in each empirical study. It is difficult to recruit potential participants from the software industry because they are usually busy. It is also difficult to arrange time with software professionals to take part in the studies during their working hours. Moreover, it could be expensive to hire software professionals for the empirical studies. Because of these constraints, it is not feasible to test all of the hypotheses within the duration of our research.

However, in this thesis we plan to test only one hypothesis. The following items and questions were considered when designing the experiment to test one of the hypotheses. These were based on the questions in the ethics application for conducting research that involved human participants at the University of Auckland.

- **Study design**: What the type of empirical study will be used to test the hypothesis? What is the method for collecting data?
- **Materials**: What materials do we need to prepare? How easy is it to prepare the materials?
- **Participants**: What is the target population? How do we recruit potential participants? How many participants do we need?
- **Tasks**: What tasks must the participants perform? Where will they perform the tasks? How easy are the tasks?
- **Time**: What is the duration of the study? Can we conduct the study within a reasonable amount of time?

Based on the questions and the limitations of our research, hypothesis 4a is the most feasible hypothesis to test. Moreover, we believe that requirements engineering is a good starting point to test the importance of transparency in software engineering. Requirements engineering is important to software development because the success of a software system depends on how well requirements satisfy the expectations of stakeholders [82]. Furthermore, the requirements engineering process consists of different communication activities which occur early in the software development life cycle. Example communication activities include interviews, brainstorming, documentations, group discussion, and requirements inspection [1, 7, 61, 82, 105]. Therefore, we choose hypothesis 4a as the first hypothesis to test if transparency can improve communication in the form of documents in requirements engineering.

We devise an experiment to compare different software artefacts. An experiment involves activities in changing variables and observing the effects caused by such changes [33, 37, 122]. Such an experiment is appropriate for this study because it provides a systematic and quantifiable way for validating theories and measures as well as evaluating the relationships between different variables [122].

The materials for the experiment are requirements documents that we can prepare within the duration of our research. We can recruit software practitioners or tertiary students (studying in software engineering-related areas) to evaluate the effectiveness of requirements documents in presenting software system requirements. Moreover, the duration of the study can be shorter because training of participants is not required. Training is unnecessary as software practitioners and tertiary students should know about the languages used in the requirements documents. They are likely to have experience with different types of software artefacts for their work or study. Participants are not required to learn another language or notation for reading a requirements document.

4.3 Summary

In this chapter we describe our research approach through three main stages: exploration, evaluation, and application. This thesis focuses on the exploration and evaluation of transparency. The application of transparency will be in the future work. At the exploration stage, we answer research questions, RQ1 and RQ2, through literature reviews and an exploratory survey. At the evaluation stage, we answer RQ3 by first deriving a set of hypotheses for the scope of the evaluation. In this thesis, we test one of the hypotheses: a transparent requirements document is more effective for developers in answering questions about the requirements of a software system than a non-transparent requirements document. We test this hypothesis through an experiment which compares the effectiveness of two types of requirements documents with different degrees of transparency.

In the following chapters we discuss how we collect supporting evidence by means of a survey and an experiment to answer our research questions. The results obtained from these two empirical methods are also presented.

5

A Survey to Explore Transparency in Software Engineering

So far we have considered different definitions of transparency (Chapter 2) and uses of transparency in software engineering (Chapter 3). To test the usefulness of our definition to software practitioners and to learn the importance of transparency in software engineering, we conducted a survey early in our research. The main purposes of the survey are to help us to gain insights into problems of communication in software projects and to evaluate our preliminary definition of transparency (Chapter 4).

The structure of this chapter is based on the reporting guidelines by Jedlitschka et al. [57]. The guidelines enable reporting of experiments in software engineering to help readers to find information, to understand how an experiment is being conducted, and to assess validity of results. We follow the guidelines when documenting our studies to ensure that sufficient information is provided to help other researchers. Any researchers who are interested in conducting similar studies can use this information to replicate and to evaluate our research.

In the following sections, we describe the design and the execution of the survey. We present our analysis of the responses collected in Section 5.3. In Section 5.4, we discuss threats to the survey's validity and summarise the findings.

5.1 Survey Design

We design the survey with a goal definition template by Wohlin et al. [122]. The purpose of the goal definition template is to help us to ensure that we have defined the important aspects of an experiment [122]. See Appendix A for our draft goal definition of the survey. We also derive the questionnaire from the questions under the purpose section of our goal definition. Each question under the purpose section has corresponding question numbers to the questionnaire in parenthesis or written in pencil as shown in Appendix A.

In the following sections, we discuss the survey design in detail. The following section begins with the main goal and research questions for the survey, then continues with describing the target population and the sampling method of the survey. The materials used and the tasks asked of the participants are presented in Section 5.1.3 and Section 5.1.4. In the last two subsections we discuss the hypotheses to be tested in the survey and the type of survey design.

5.1.1 Goal

The main goal of the survey is to gather evidence to answer RQ2: what is transparency in the software engineering context? To answer that question, the following questions are addressed in the survey:

- **SQ1.** What are the communication problems encountered by stakeholders of a software project?

  In Chapter 1, we discussed how transparency could be useful for addressing communication problems in software life cycle. We construct SQ1 to further explore different types of communication problems that stakeholders might encounter in software projects. We test whether the communication problems reported by stakeholders are related to the three attributes of transparency: accessibility, understandability, and relevance.

- **SQ2.** What does "transparency" mean to different stakeholders of a software project?

  In Chapter 2 and Chapter 3, we found that different professions used the term "transparency" for various purposes. However, the term has not been well defined in the software engineering literature. We want SQ2 to explore what transparency means to different stakeholders. Specifically, we want to know if stakeholders are familiar with the term "transparency" and what definitions of transparency they might already be familiar with. We are also interested in stakeholders' opinions of our definition of transparency, especially if our definition is consistent with the one that they might already have.

5.1.2 Participants

In this subsection, we describe the type of participants that we aim to recruit for the survey. We also describe the sampling method for the survey.

Target Population

The target population for the survey is software project stakeholders such as requirements engineers, software developers, project managers, clients, end users, and government regulators. We are interested in stakeholders' communication problems and what transparency means to them. We aim to recruit anyone who is or has been involved in software projects as in the list at the start of this section. Survey responses from different types of stakeholders enable us to explore different types of communication problems in software projects. We are also exploring different perspectives on transparency from stakeholders.

Sampling Method

We use convenience sampling as the procedure for selecting potential participants. Any potential participants are stakeholders who are available and willing to participate. Convenience sampling is used because we lack sufficient information about the entire population involved in software projects. It is also difficult to recruit everyone involved in software projects. Convenience sampling helps us to recruit participants who are readily available and to save time in finding potential participants.

5.1.3 Survey Material

A web-based questionnaire to gather responses from participants saves time in distributing the survey to participants. It also saves time in manually entering data. Moreover, it simplifies the process for participants to answer the survey [89].

The web-based questionnaire contains 28 questions in total and is divided into three main sections. See Appendix C for the questionnaire.

1. Demographics (Q1 – Q3)

   The first section of the questionnaire asks participants what aspects of a software project that they are involved in and what their roles are in a software project. We also ask participants to rate their knowledge or experience in several areas such as requirements engineering. We use the responses collected from this section to classify participants.

2. Communication in Software Development (Q4 – Q14)

   In this section, we explore what kind of communication problems that our participants encountered in a software project. The responses collected from this section answer SQ1. Participants are asked to report the frequency of communication problems and the types of information that our participants need in a software project. We ask participants how they obtain necessary information and how they communicate with other stakeholders. The questions constructed in this section are based on the roles of senders and receivers in the simple communication model.

3. Transparency in Software Engineering (Q15 – Q26)

   This section is constructed to explore the meaning of "transparency" to participants for answering SQ2. In this section, we evaluate our preliminary definition and the three attributes (accessibility, understandability, and relevance) for transparency. We first ask participants to select contexts in which transparency appears that they are familiar with. We then present our definition and ask participants if they are aware of our definition within software engineering. With our attributes of transparency, participants are asked to rate the importance of each attribute to our definition of transparency as well as the effectiveness of different techniques in making information transparent.

At the end of the questionnaire, we have two open questions (Q27 – Q28). In Q27, participants can comment on their concerns about the concept of transparency. Participants can also comment on any other problems that they are concerned within software engineering in Q28.

5.1.4 Tasks

The survey asks participants to look back at their experience in software projects. The main task asked of participants is to answer the web-based questionnaire, which takes approximately 30 minutes.

5.1.5 Survey Hypotheses

In this subsection we describe the hypotheses to be tested in the survey. We also describe the variables in the survey.

Hypotheses

We formulate the following hypotheses from SQ1 and SQ2.

> **SH1.** A majority of stakeholders frequently or always encounter communication problems relevant to transparency problems.

We discuss transparency as a useful concept for improving communication in the software life cycle in Chapter 1. If transparency is important to communication, many communication problems encountered by stakeholders would be related to transparency problems. We hypothesise that at least 50% of the participants report that they frequently or always encounter one or more of our designated transparency problems when they receive or send information. The designated transparency problems are inaccessibility, misunderstanding, and irrelevance of information.

> **SH2.** A majority of stakeholders of a software project are familiar with the term "transparency" used in more than one context.

SH2 enables us to see whether many stakeholders are familiar with different definitions of transparency. If stakeholders are familiar with transparency, we are interested in finding what kind of definitions stakeholders know. We want to know if our definition is consistent with the definitions that stakeholders have. We hypothesise that at least 50% of the participants are familiar with transparency used in more than one context.

Variables

According to Wohlin et al. [122], there are two types of variables in an experiment, independent variables and dependent variables. The independent variables are the input variables, where the experimenter applies different treatments to these variables. The dependent variables are the outcomes of the effect of the changes in the independent variables [122].

In the survey it is not possible to apply any treatments to the independent variables because we do not have control over the variables. We can only observe the differences in the variables. The following three independent variables relate to SH1 and SH2:

- The roles that participants have in a software project;

- The level of knowledge in software engineering that participants have;
- The experience in software development that participants have.

We are interested in participants' roles and knowledge because they come from different backgrounds and have different experience with transparency. For example, developers might be familiar with transparency in networking; on the other hand, clients might be familiar with transparency in governments and public participation. We are also interested in participants' experience as participants with different levels of expertise might face different communication problems.

The dependent variables for both SH1 and SH2 are the participants' responses to the survey. In particular, we are interested in the frequencies of communication problems reported by participants for testing SH1. To test SH2, we consider the different definitions of transparency selected by participants.

5.1.6 Design

In this subsection, we describe the type of design for the survey. We also discuss the ethical issues that arise from the survey in the following subsections.

Type of Survey Design

The type of survey design is cross-sectional which asks participants information at one fixed point in time [59]. The survey is also a retrospective study. In the survey we ask participants to provide information about their experience in software projects as well as their knowledge about transparency. The web-based questionnaire used for the survey is self-administered, participants can complete the questionnaire on their own.

Ethical Considerations

At the University of Auckland, it is required to apply to the University's Human Participants Ethics Committee (UAHPEC) for conducting research that involves human participants. Ethics approval is important because we need to ensure that our research does not harm participants in any way. Following ethical issues arise from the survey, and the survey addressed them:

1. Anonymity

   To protect participants' privacy, we do not ask for information that might identify individual participants in the survey. We also avoid questions that could potentially reveal information about other individuals or organisations. However, it is possible that participants accidentally reveal personal or organisational information. To minimise the possibilities, we ask participants not to provide any identifying information in their responses. All identifying information disclosed from the responses is removed. In addition, the information provided by the participants is analysed and reported anonymously.

Since the survey involves the use of a web-based questionnaire via SurveyMonkey (an on-line survey provider), it is possible to identify participants and their locations by their IP addresses. To protect participants' anonymity, we do not record IP addresses nor do we ask participants for their email addresses or any information that directly reveals their identities.

2. Confidentiality

   To protect confidentiality of participants' answers, we do not make data available to the public. We remove any identifying information in the responses. Since the survey is designed to be anonymous, we have no information intentionally given about the identity of participants.

3. Rights to withdraw

   It is not possible for participants to withdraw data from our research after they submit the web-based questionnaire. This is because the anonymous responses make it impossible to identify any specific completed survey. Participants are made aware that they cannot withdraw data after clicking the submit button at the end of the web-based questionnaire. However, participants are entitled to withdraw from involvement in the survey at any time before submitting the questionnaire.

4. Informed consent

   Participants are not required to sign consent forms for the survey because the collection of the survey is anonymous. However, we include a consent page at the beginning of the web-based questionnaire. This informs potential participants about the survey and ensures that they understand what is involved in the survey. The consent page also informs participants about their rights to withdraw from our research.

5.2 Execution

In this section the execution of the survey involves what we prepared for conducting the survey. This section also gives an overview of the web-based questionnaire procedure.

5.2.1 Preparation

To conduct the survey, we applied for ethics approval to the UAHPEC. We received the approval from the ethics committee in September 2010 (see Appendix B for our application submitted to the Ethics Committee).

To recruit potential participants from software industry, we emailed invitations for participation to 40 software engineering graduates and several software professionals in New Zealand. We chose software engineering graduates and software professionals because they were most likely to be involved in software projects as developers or project managers. To recruit other types of stakeholders such as user representatives or clients, we also obtained permission to forward the invitations to a mailing list in the Faculty of Medical and Health Sciences at the University of Auckland.

In the invitation email, we included a participant information sheet (PIS) which described the purpose of the survey and the types of participants we sought. The PIS also contained information about data storage, anonymity of responses as well as participants' rights to withdraw from our research. We also provided a link to our survey web page in both the PIS and the invitation email. Participation was entirely voluntary. Participants were not required to sign consent forms because the survey was designed to be anonymous. See Appendix B for the PIS and the invitation email.

5.2.2 Procedure

The survey was run between October 2010 and March 2011. We provided a link to the web-based questionnaire for potential participants in the invitation email and the PIS. Potential participants could answer the questionnaire at any time. Figure 5.1 illustrates a screen shot of one part of the web-based questionnaire.

To start the web-based questionnaire, a consent page was presented to participants. The main purpose of the consent page was to ensure that the participants understood the conditions for taking part in the survey. The consent page only asked the participants to tick "agree" or "disagree" to participation. Participants were not asked to sign their names or to put any information that could be used to identify them on the consent page. The questionnaire would proceed when the participants chose "agree" from the consent page.

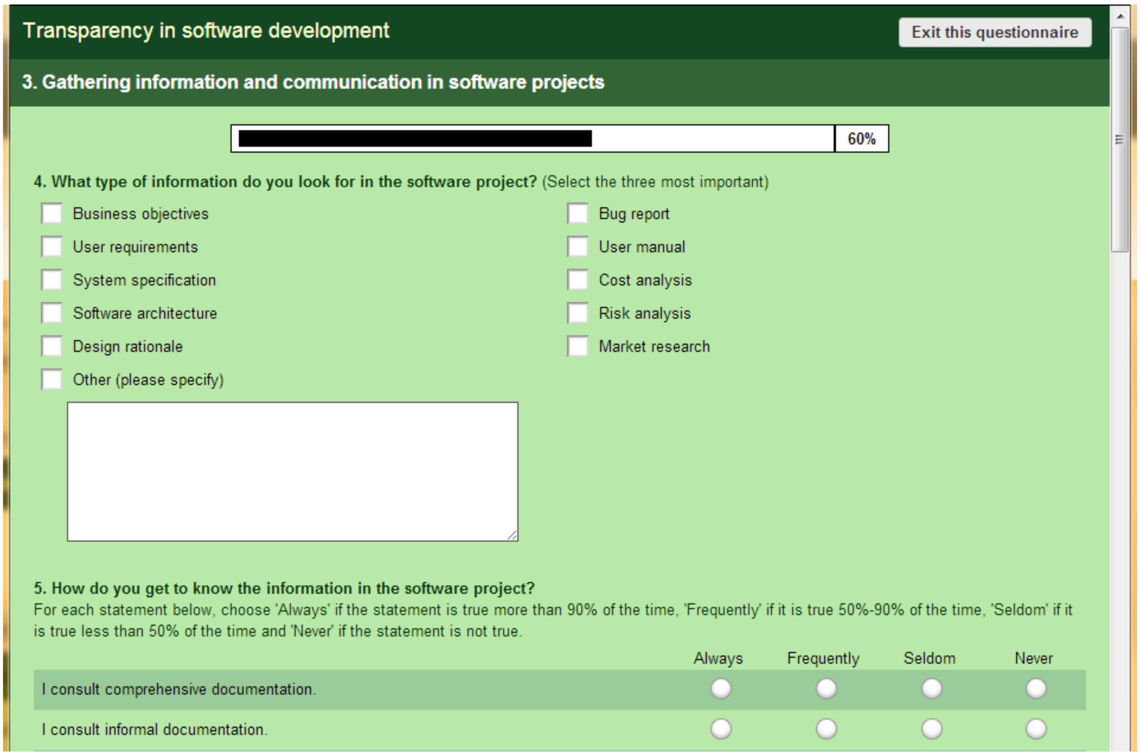


Figure 5.1: A part of the web-based questionnaire used in the survey.

The web-based questionnaire was divided into the following web pages:

1. Participants Consent Form.
2. Demographics (Q1 – Q3).
3. Gathering information and communication in software projects (Q4 – Q14).
4. Transparency in software engineering (Q15 – Q26) and Overall comments (Q27 – Q28).
5. Thank you.

Participants were not required to answer all of the questions. They progressed through the questionnaire using the "Next" and "Prev" buttons. When participants reached the last page of the questionnaire (the thank you page), a "Submit" button appeared. Participants were asked to click the "Submit" button to complete the questionnaire. Any responses made without clicking the "Submit" button were treated as withdrawing from participating in the research. These responses were not used in the research.

5.3 Analysis

By March 2011, we received 21 complete responses. We considered complete responses as those that completed more than 80% of the questions and clicked the submit button at the end of the questionnaire. There were 43 partial responses and complete responses in total. Out of the 43 responses, only 22 clicked the submit button to complete the questionnaire. We removed 1 response from the 22 responses because the response only completed the first demographic question.

From the 21 complete responses, most participants did not answer the two open questions (Q27, Q28) at the end of the questionnaire. 18 participants answered the first 26 questions. There were three participants who did not answer some of the questions. One participant missed Q12 and Q26. Q12 asked participants to report the frequency of communication problems that they encountered when they were communicating with other stakeholders. Q26 asked participants to rate the effectiveness of different techniques in making information accessible, relevant, or understandable to stakeholders. One participant did not answer Q11 which asked participants to rate how helpful they found different approaches were in helping them to communicate with other stakeholders. The other participant did not answer Q21 which asked participants about the types of communication problems related to our definition of transparency.

To perform statistical analysis, we transformed the frequency scale (Never, Seldom, Frequently, and Always) into numerical values (0, 1, 2, and 3). Similarly, we changed the five-point scale of "Very poor, Poor, Average, Good and Very Good" into numerical values of 0, 1, 2, 3, and 4. Thus, we used the transformed values to count the frequency and to test the two hypotheses. We also assigned values for missing responses in Q11, Q12, Q21 and Q26. We assumed that the questions were not applicable to the participants. For example, in Q11 we asked participants to rate the effectiveness of each method/technique for helping participants to communicate with other stakeholders. We assigned "N/A" as the value for each method/technique listed in Q11.

The following sections present the statistical analysis of the 21 complete responses gathered from the survey. We organise the analysis based on the structure of the questionnaire. In the last two sections, we present the results for testing the two hypotheses, SH1 and SH2.

5.3.1 Demographics

Of the 21 complete responses, 20 participants (95.2%) were involved in software development, followed by software testing (66.7%) and software design (61.9%). These percentages show that most participants were involved in more than one aspect of a software project. Figure 5.2 illustrates the number of participants involved in each aspect of a software project. One participant also reported that he or she was a client for several projects.

Most participants also have more than one role in a software project. Figure 5.3 shows the number of participants involved in each role of a software project. Most participants have roles as developer (81%), requirements engineer (38.8%), and architect (33.3%). Two participants also reported in the other category that they have the role of “test analyst” or “tester”.

In the questionnaire, participants were asked to assess their knowledge or experience in the following areas: the software project; requirements engineering; communicating with different stakeholders; and software engineering.

On a five-point scale (Very poor, Poor, Average, Good, Very good), 16 participants reported that they have good or very good knowledge of the software project. More than half of the participants reported that they have good or very good knowledge of requirements engineering. Similarly, more than half of the participants said they have good or very good knowledge of software engineering. Out of the 21 participants, 17 participants also reported that they were good or very good at communicating with different stakeholders. No participants rated themselves as being poor or very poor at communication. Figure 5.4 illustrates participants’ self-assessed knowledge or experience on a five-point scale in the four areas specified.

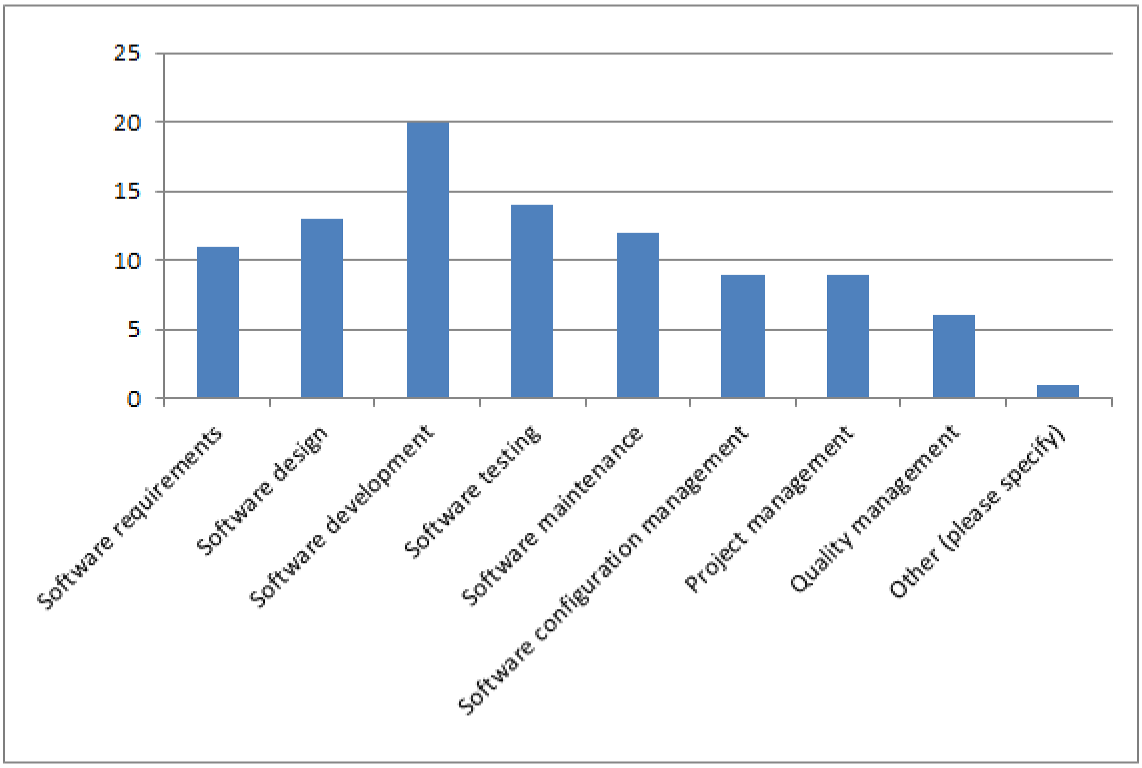


Figure 5.2: Number of participants involved in each aspect of a software project.

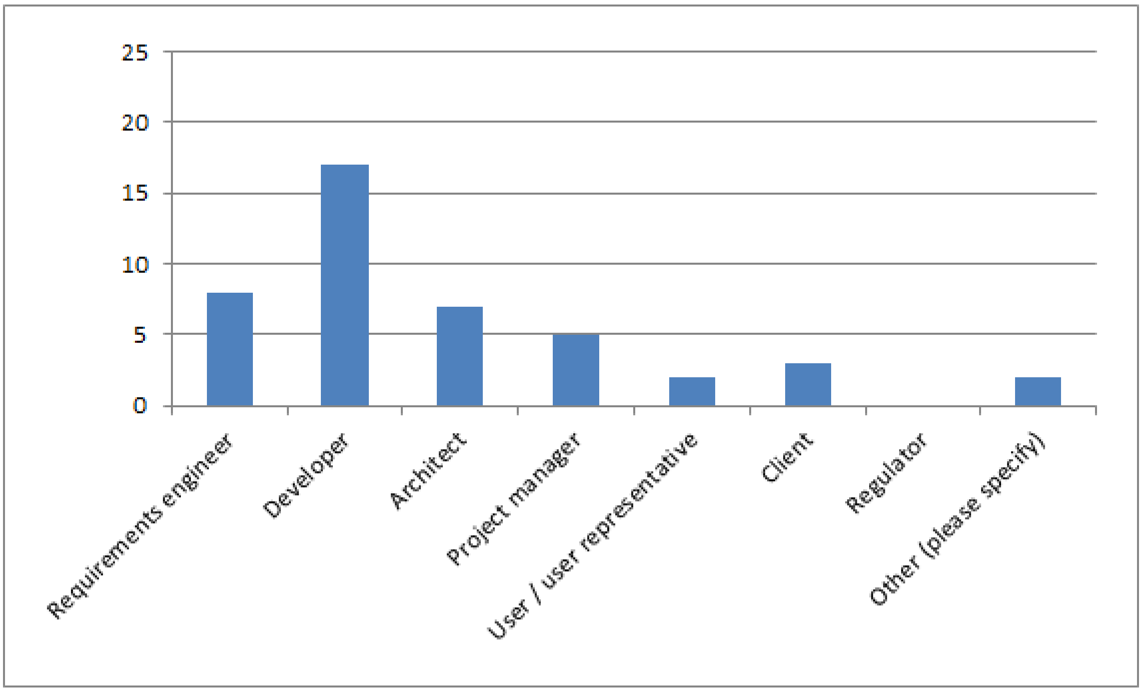


Figure 5.3: Number of participants involved in each role of a software project.

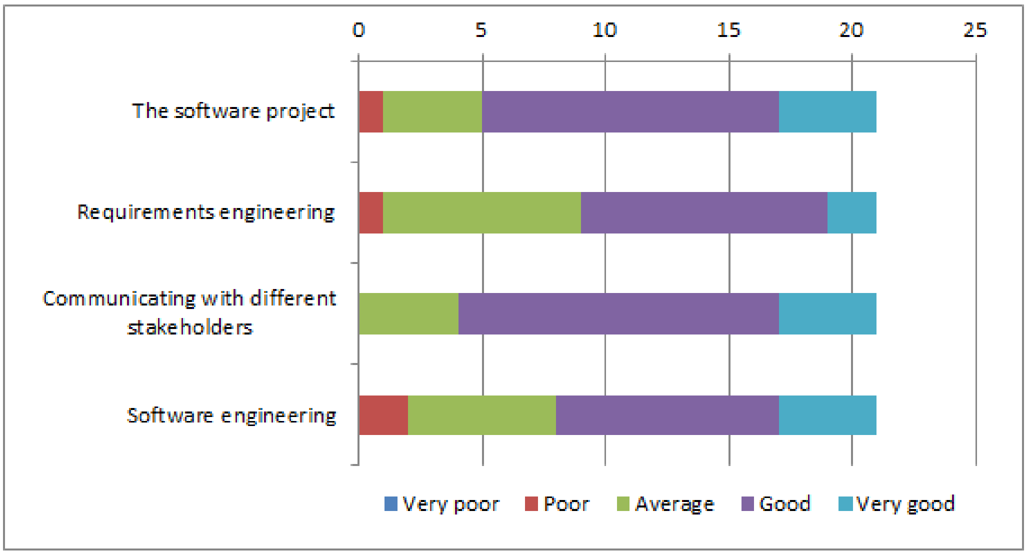


Figure 5.4: Participants' self-assessments on how well they believe that their knowledge or experience is in the areas specified.

5.3.2 Communication Problems in Software Development

We used three general questions to explore communication in software projects. The three questions were then refined based on the roles of senders and receivers of the communication model. In the following sections, we present the results of these questions.

What information do stakeholders use in a software project?

When the participants were receivers of the communication model, they sought user requirements as the type of information (85.7%). The second type of information was business objectives (81%) and followed by design rationale (57.1%). When the participants were senders, the type of information that most participants used to convey to other stakeholders was user requirements (66.7%). This was followed by design rationale (61.9%) and system specification (47.6%). The most common type of stakeholders that the participants communicated with was project manager (81%), developer (66.7%), client (66.7%) and user/user representative (52.4%).

How do stakeholders communicate in a software project?

As illustrated in Figure 5.5, we asked our participants in Q5 of the questionnaire: "How do you get to know the information in the software project?" This question enabled us to explore ways receivers of the communication model used to receive information. The top three ways that most participants reported frequently or always used are:

1. “I learn about the information by informal discussion with other members of my organisation”

2. “I consult informal documentation”

3. “I have to search for the information that I need”

Figure 5.6 illustrates the effectiveness of each way for getting to know information in a software project. “I learn about the information by informal discussion with other members of my organisation”, “I have to search for the information that I need”, “I consult informal documentation” and “I learn about the information at planning meetings” were the top ways that most participants rated good or very good.

To explore how senders communicate with receivers, we asked our participants how they communicated with other stakeholders about a software project. Figure 5.7 shows the different usages by the participants to communicate with other stakeholders. The top three ways that most participants reported frequently or always used are:

1. “I give the information to other stakeholders at planning meetings”

2. “I give information about the project by informal discussions with other members of my organisation”

3. “I ask other stakeholders to consult informal documentation”

We also asked our participants to rate the effectiveness of each way for communicating with other stakeholders, shown in Figure 5.8. The top ways that most participants rated good or very good are:

1. “I give information about the project by informal discussion with other members of my organisation”

2. “I give the information to other stakeholders at planning meetings”

3. “I give information about the project by informal discussion with clients”

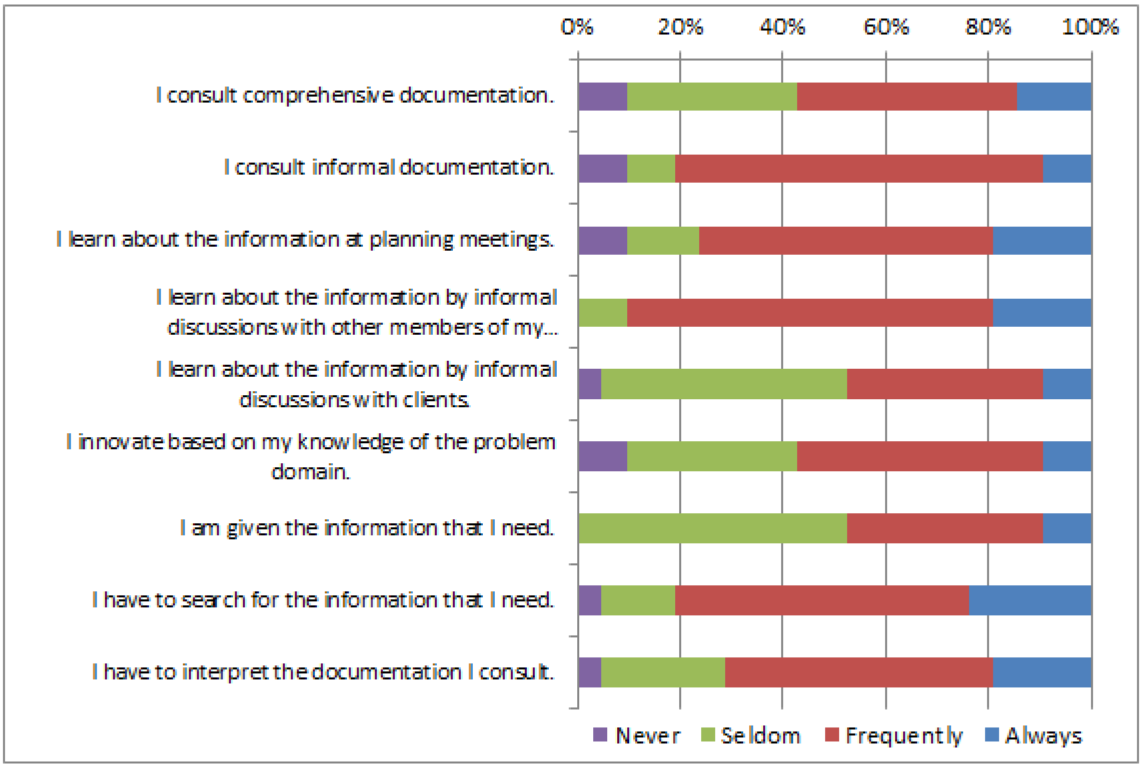


Figure 5.5: Frequency of using different ways for getting to know information in a software project.

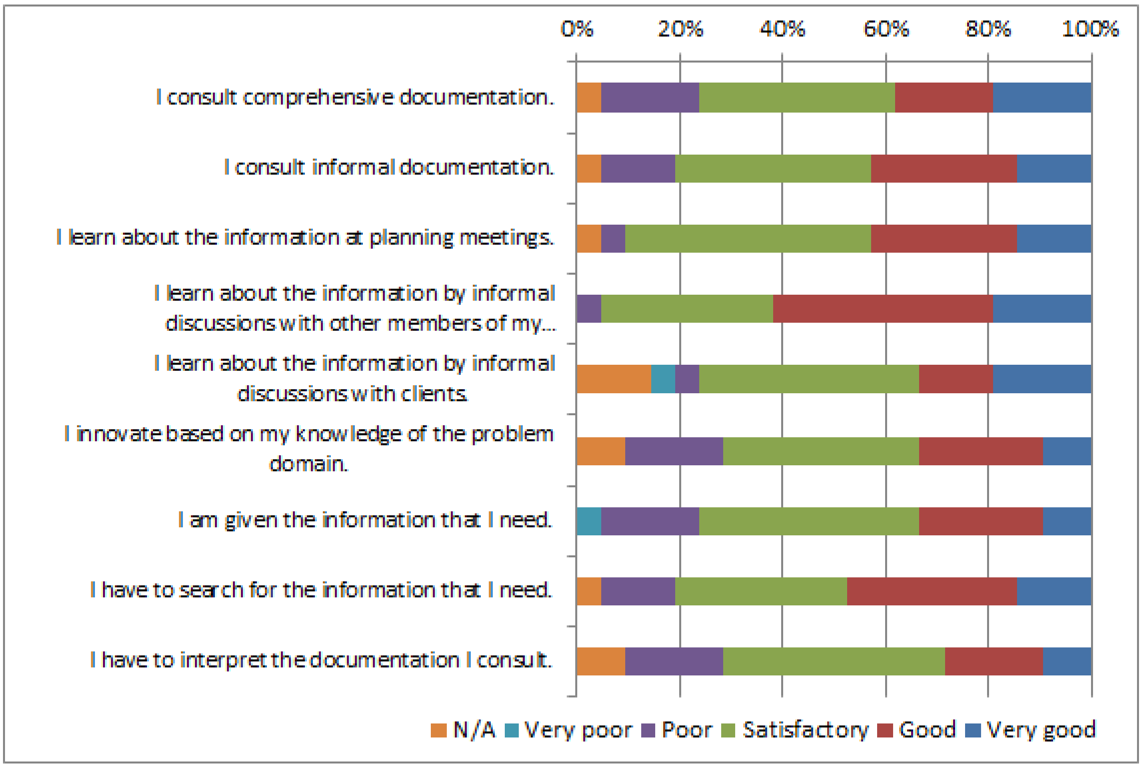


Figure 5.6: Effectiveness of different ways for getting to know information in a software project.

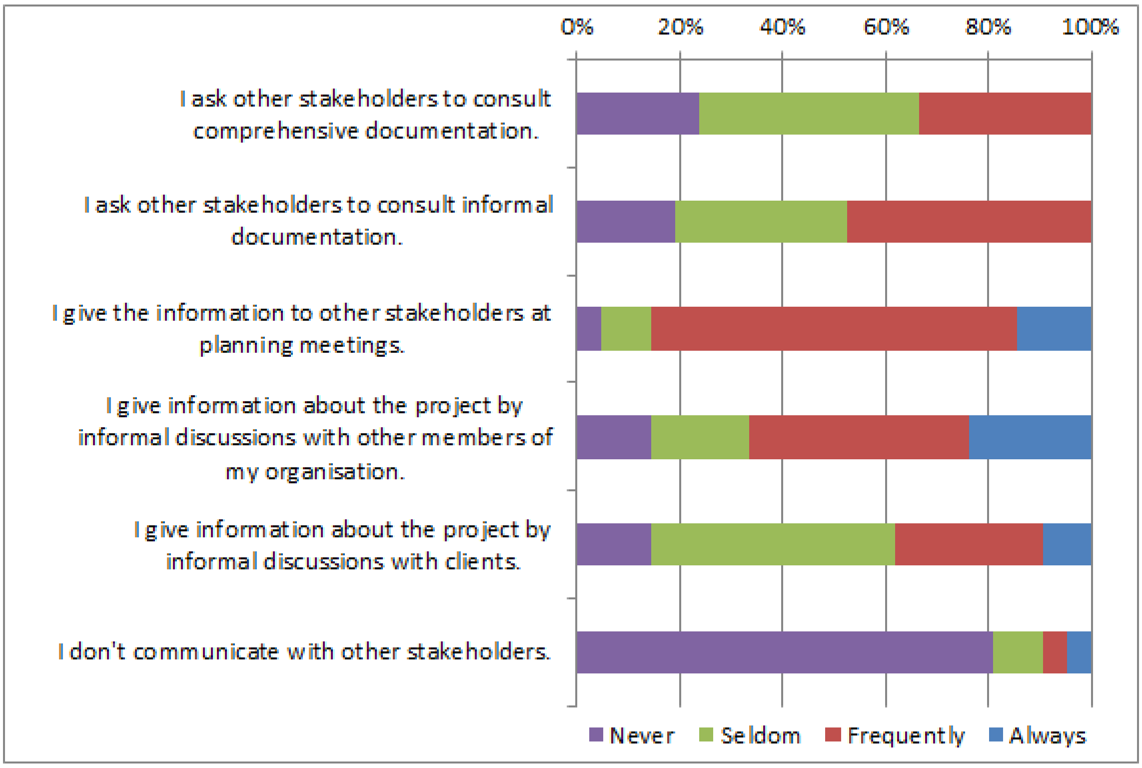


Figure 5.7: Frequency of using different ways for communicating with other stakeholders about a software project.

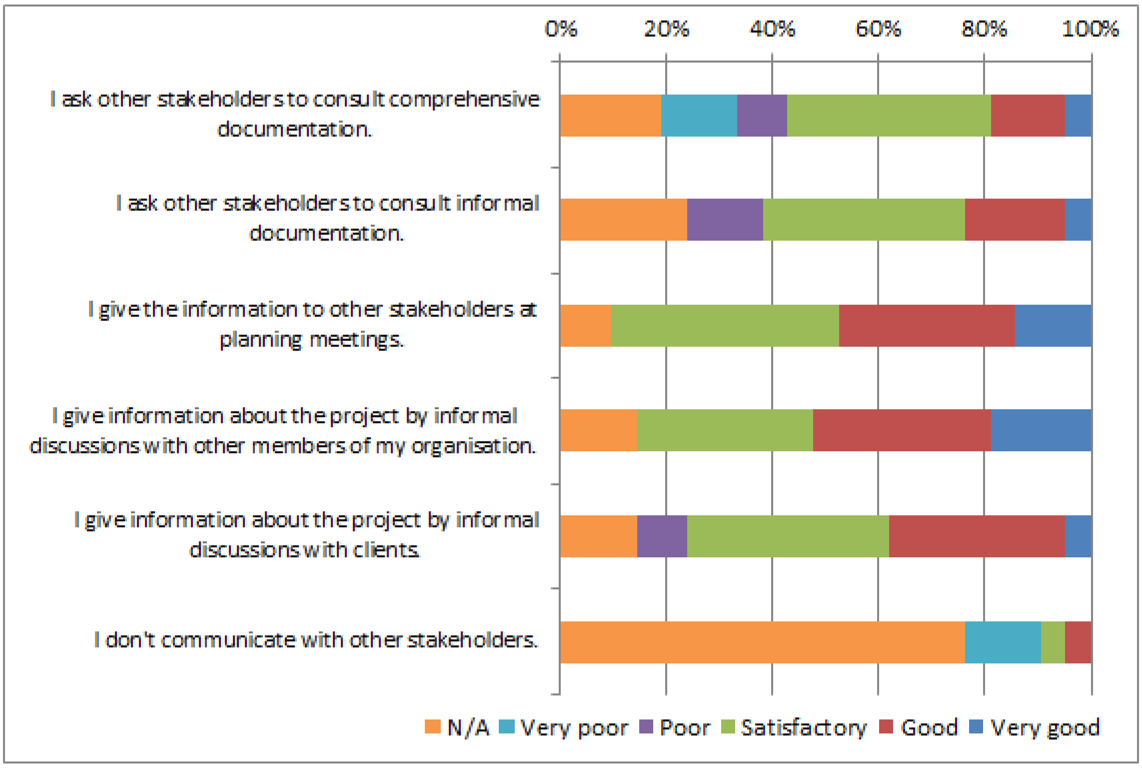


Figure 5.8: Effectiveness of different ways for communicating with other stakeholders about a software project.

Q7. What problems do you encounter when trying to know the information in the software project?

a) There are too many managers and clients to deal with.

b) The information is difficult to understand.

c) I don't know what information to look for in the software project.

d) I can't find the information or it is difficult to obtain the information that I need.

e) The information contains errors.

f) The given information is not what I need.

Q12. What problems do you encounter when you communicate with other stakeholders?

a) There are too many managers and clients to deal with.

b) The information is difficult to understand for other stakeholders.

c) I don't know what information to give to other stakeholders.

d) Other stakeholders can't find the information or it is difficult to obtain the information.

e) The information contains errors.

f) The information given to the stakeholders is not what they need.

Questionnaire Text 1: Q7 and Q12 from the questionnaire for exploring the types of communication problems encountered by our participants.

What are the communication problems in a software project that stakeholders encountered?

To explore the types of communication problems encountered by receivers and senders of the communication model, we constructed Q7 (receiver) and Q12 (sender) in the questionnaire. There were six types of problems listed in Q7 and Q12 as shown in Questionnaire Text 1. We summarised each problem from Q7 and Q12 into one word, which can be found in Table 5.1. Participants were asked to report the frequency of each problem occurring in the software project.

Q7 is constructed to help us to find the types of problems that the participants encountered as receivers of the communication model. Figure 5.9 illustrates the frequency of problems in getting to know information reported by the participants. In addition to the problems listed in Q7, one participant commented on "certain problems are not mentioned in the documentation // Frequently". The top three problems that most participants reported frequently or always were:

Question Number	Type of Problem
Q7a, Q12a	Manageability
Q7b, Q12b	Understandability
Q7c, Q12c	Competency
Q7d, Q12d	Accessibility
Q7e, Q12e	Accuracy
Q7f, Q12f	Relevance

Table 5.1: Each type of problem listed in Q7 and Q12 of the questionnaire and its one-word descriptor.

1. "I can't find the information or it is difficult to obtain the information" (accessibility).
2. "The given information is not what I need" (relevance).
3. "The information contains errors" (accuracy).

When the participants were senders, the types of communication problems encountered by the participants were slightly different. Figure 5.10 illustrates the frequency of problems in communicating with other stakeholders reported by the participants. The problems that most participants reported frequently or always encountered were:

1. "The information is difficult to understand for other stakeholders" (understandability).
2. "Other stakeholders can't find the information or it is difficult to obtain the information" (accessibility).
3. "The information contains errors" (accuracy).

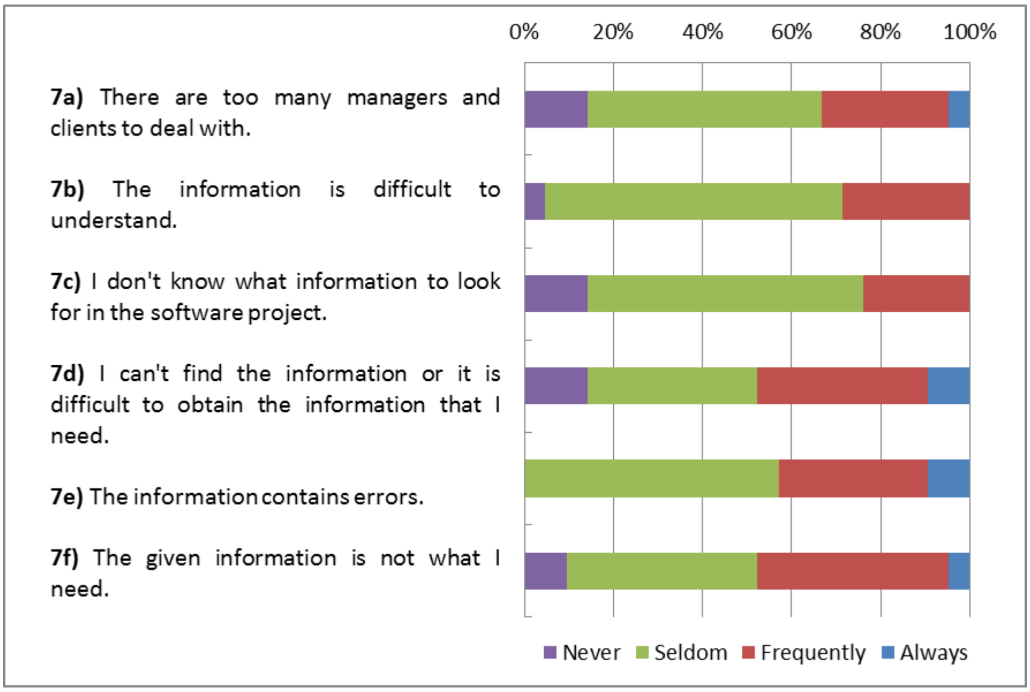


Figure 5.9: Frequency of each communication problem occurring in a software project when participants were receivers of information.

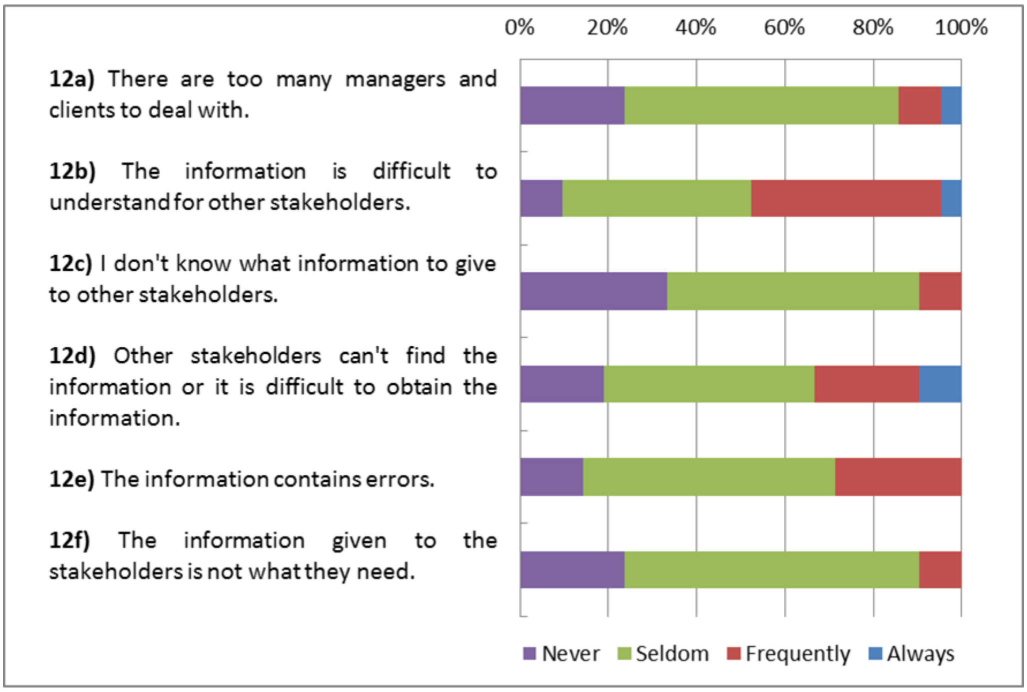


Figure 5.10: Frequency of each communication problem occurring in a software project when participants were senders of information.

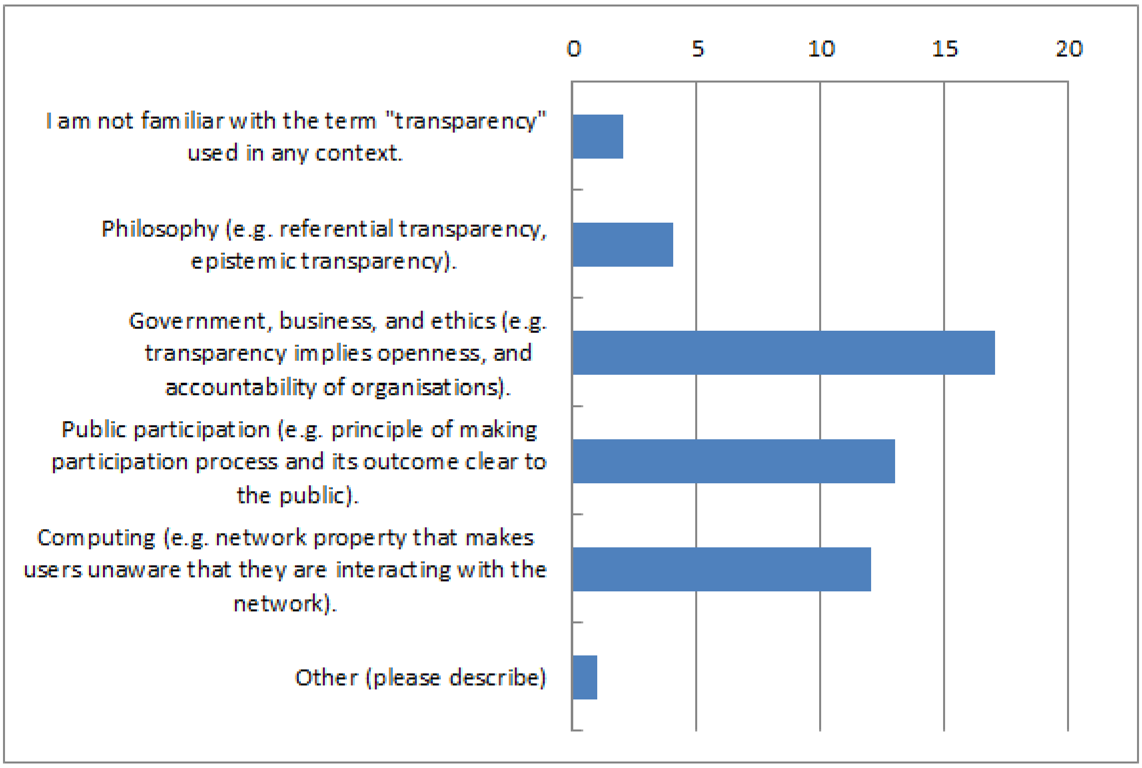


Figure 5.11: Number of participants who were familiar with the term "transparency" used in each context.

5.3.3 Transparency in Software Development

In the third part of the survey, we asked participants questions about the term "transparency" used in different contexts. We also asked participants questions about our preliminary definition of transparency. In the following sections we summarise the responses to these questions.

What does transparency mean to stakeholders?

As shown in Figure 5.11, more than 50% of the participants were familiar with transparency used in different contexts. In particular, 17 out of 21 participants were familiar with transparency in the context of government, business and ethics. Out of 21 participants, 13 participants were also familiar with transparency used in public participation. In addition, we received one response which referred to transparency in the context of physics as "light being able to pass through material".

We define transparency in software engineering as:

> **Enabling stakeholders to answer their questions about the software project.**

A stakeholder can be anyone involved in the software project, e.g. users, market analysts, software engineers.

Questionnaire Text 2: Our preliminary definition of transparency presented to participants in the survey.

How do stakeholders perceive our definition of transparency?

Our preliminary definition of transparency was presented to the participants in the questionnaire as shown in Questionnaire Text 2. About half of the participants did not know if our definition had been considered in software engineering. More than half of the participants indicated that our definition was important to help them to know about the software project and to communicate with other stakeholders (76.2% and 81.0% respectively). Moreover, one participant mentioned that transparency "allows for efficient progress of the project as all the stakeholders are aware of their responsibilities in the project and the various dependencies that might exist within the project". However, one participant commented that he or she did not understand our definition.

We also asked the participants if any terms other than transparency were used to describe our definition in software engineering. Four participants answered no other term was used to describe our definition in software engineering. Of the 21 participants, 14 participants did not know if any other terms were used. The remaining participants answered the terms used in software engineering for our definition were "modelling", "open communication" and "communication".

Participants were asked to select the types of stakeholders that they thought required our definition. As illustrated in Figure 5.12, the types of stakeholders that most participants indicated were developer (85.7%) and client (85.7%). The type of stakeholders that fewest participants indicated was regulator (47.6%). Two participants also commented on other types of stakeholders who required transparency. One participant mentioned the client's IT management team. The participant mentioned that only those in charge of maintaining the software project needed to know about the whole project. The other participant commented that ideally all the stakeholders required transparency.

In addition, the survey contained questions about which of the communication problems listed in Q7 and Q12 were also related to our definition of transparency. Out of the 21 participants, 17 participants indicated that the accessibility problem from Q7 was related to our definition. This was followed by understandability, accuracy, competency

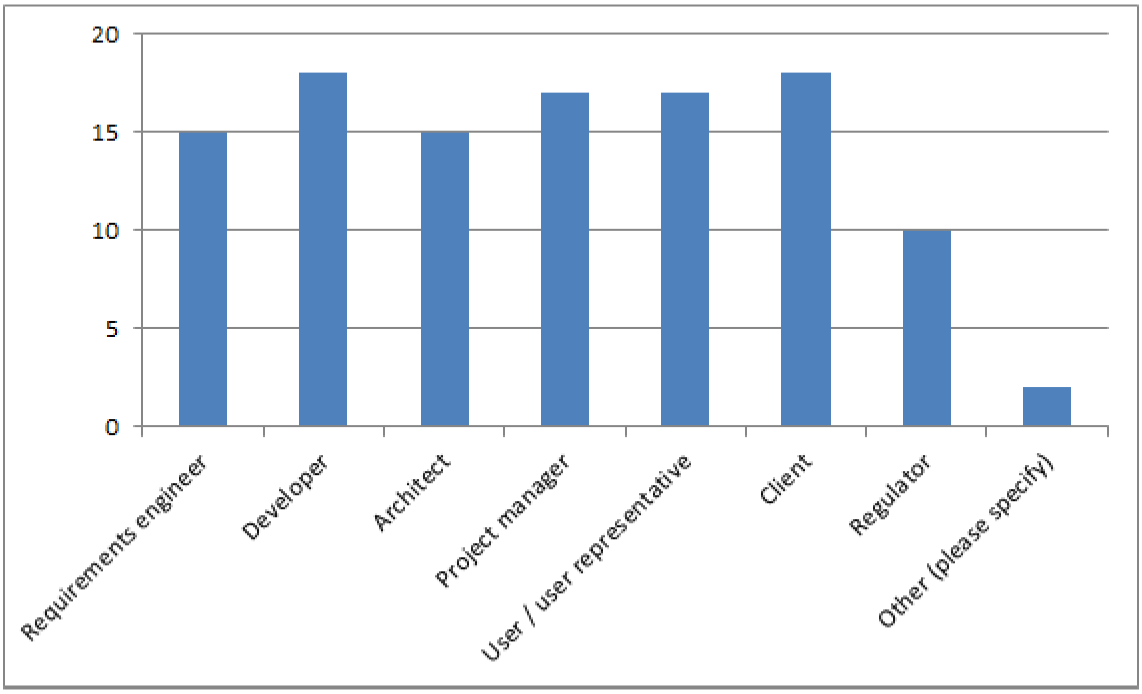


Figure 5.12: Types of stakeholders that required transparency as reported by our participants.

and relevance problems. Similarly, 16 participants indicated that the accessibility problem from Q12 was related to our definition. This was followed by understandability, relevance, accuracy, and competency problems. Figure 5.13 and Figure 5.14 show the problems listed in Q7 and Q12 related to our definition as reported by the participants.

Participants commented on other communication problems that they thought were related to our definition. When trying to know the information in the software project, one participant described the difficulty of getting information from business analysts. Another participant commented on the difficulty of setting a question when some stakeholders such as users did not have sufficient background in software engineering. When communicating with other stakeholders, one participant commented that stakeholders "understand the information given in their own ways". Furthermore, four participants commented on other problems in software engineering that were related to our definition. The problems noted by the participants were "insufficient feedback from stakeholders. . . ", ". . . [stakeholders were] not being able to completely understand the given information", ". . . lack of accountability", and "deliverables do not match requirements, needs". At the end of Q22 of the survey, we presented the three attributes of transparency as shown in Questionnaire Text 3. We asked the participants to rate the importance of the three attributes to our definition. All of the participants rated all three attributes important or very important

We believe that in order to achieve [our concept] of transparency, the information presented in software projects should have the following attributes:

- **Accessibility.** Information is accessible when it can be obtained easily.
- **Relevance.** Information is relevant when it is appropriate to the expectations of the stakeholders.
- **Understandability.** Information is understandable when it can be perceived by any stakeholders with reasonable knowledge.

Questionnaire Text 3: Our preliminary definitions for the three attributes of transparency presented to participants in the survey.

to our definition.

In the responses collected, four participants commented on other attributes that would be important to our definition. "Accuracy" was an important attribute to two participants. One participant commented that "information must be valid and up-to-date. Information must be tailored to the scope of requirements in order not to overwhelm the stakeholders". Another participant mentioned "visual, diagrams representing enterprise/organisation, information flow etc..." were important to transparency.

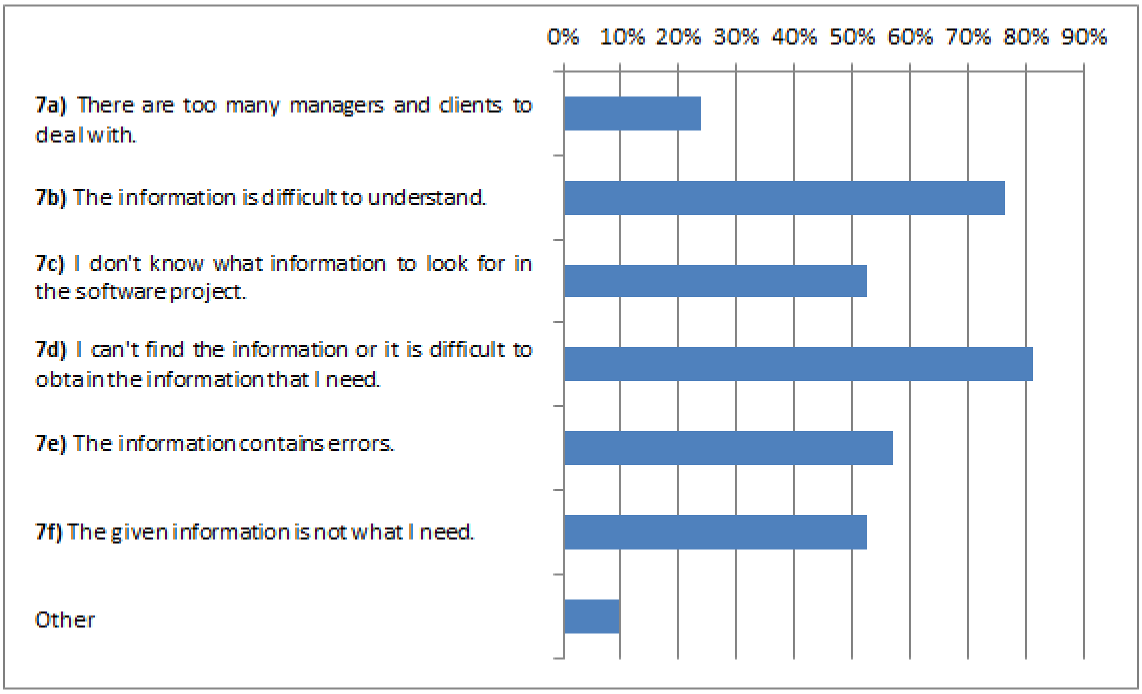


Figure 5.13: Types of problems listed in Q7 that were related to our definition of transparency as reported by our participants.

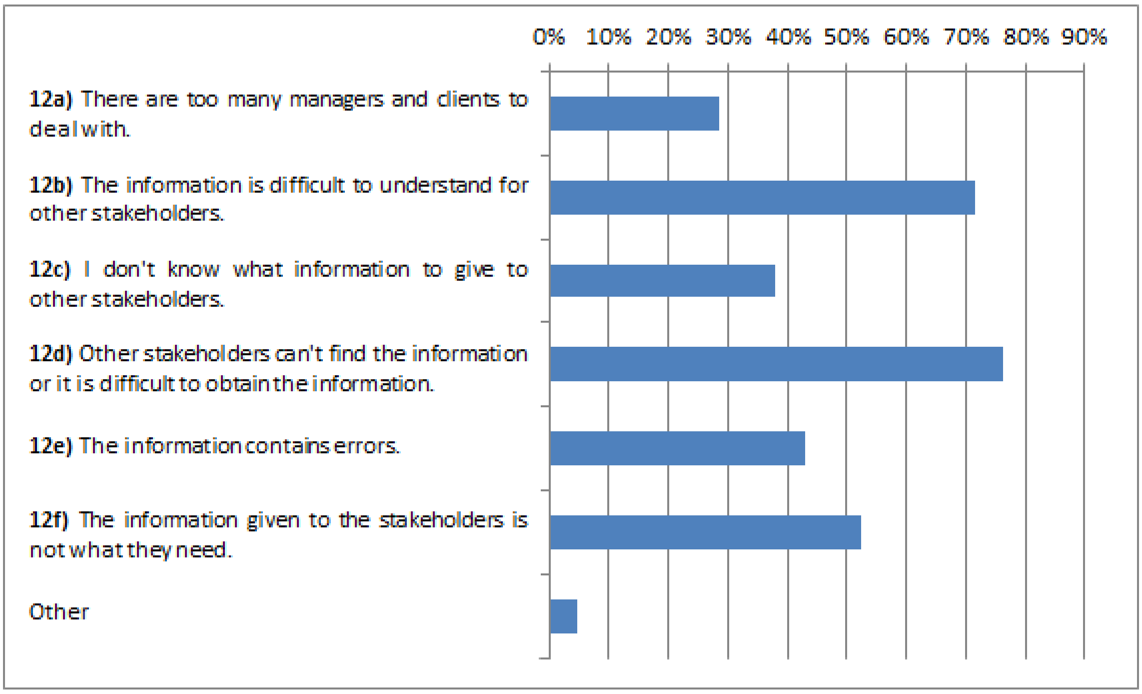


Figure 5.14: Types of problems listed in Q12 that were related to our definition of transparency as reported by our participants.

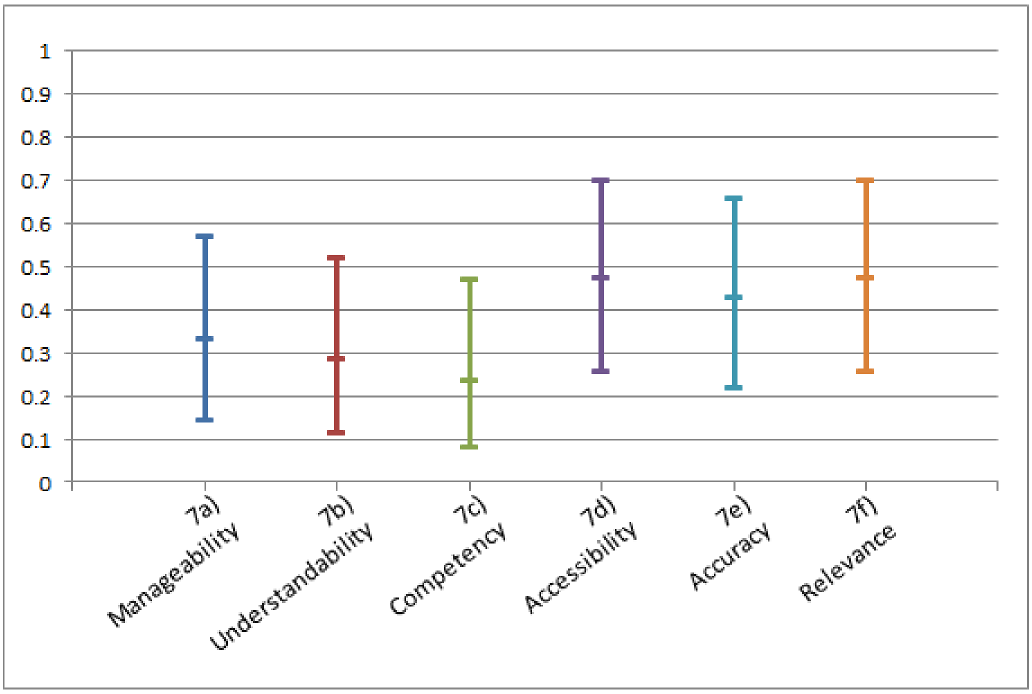
5.3.4 Transparency and Communication Problems

In order to test SH1, whether a majority of stakeholders frequently or always encounter transparency problems, we focus on Q7 and Q12 of the survey. We perform statistical analysis of the frequency data by first dividing the Likert scale into two categories: "Never or Seldom" (N + S) and "Frequently or Always" (F + A). The values assigned to the two categories are 0 and 1 respectively. A frequency count is performed to determine the number of "frequently or always" participants who had any of the transparency problems of accessibility, understandability or relevance specified in Q7 and Q12. The data show that 17 participants reported that they "frequently or always" encounter one or more transparency problems and four participants reported that they "never or seldom" encounter any transparency problems. The observed proportion of (F + A) to the observed proportion of (N + S) is 0.81:0.19.

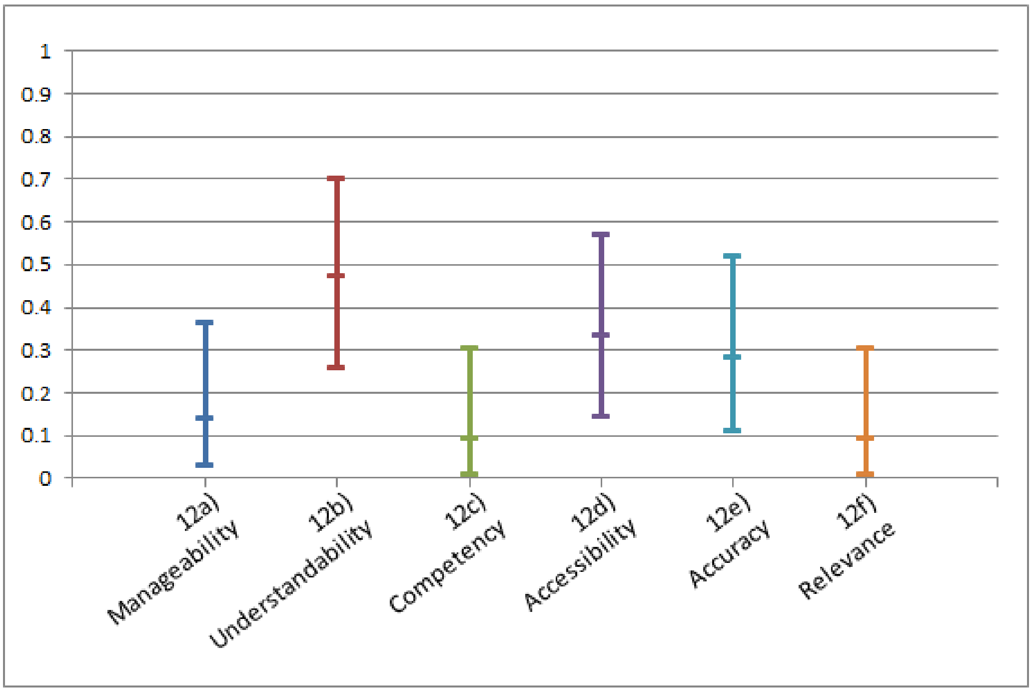
We hypothesise that 50% of the "never or seldom" participants had any transparency problems. Given the hypothesised proportion of (N + S) = 0.5, a binomial test reveals the probability of finding four or fewer "never or seldom" participants who had any transparency problems is 0.004. This probability is less than the 0.05 level of significance. The binomial test gives us high confidence that the proportion of (N + S) is less than 0.5. The proportion of (F + A) is at least 0.5, which supports our hypothesis SH1.

We then examine whether some types of communication problems occur more frequently than others. Two 95% confidence interval graphs are plotted as shown in Figure 5.15. It seems that participants have more problems with accessibility and understandability when they were receivers of information. However, there is insufficient evidence in Figure 5.15a to conclude that there is any significant difference between the occurrence of problems. This is because the confidence intervals for the occurrence of problems overlap each other. It seems that participants have problems with understandability more frequently than problems relating to competency and relevance when they were senders of information. There is a small overlap between the confidence intervals for understandability problems and problems with competency and relevance as shown in Figure 5.15b.

We also examine any differences in the occurrence of communication problems when our participants are receivers or senders of information. We take the most frequent communication problem as reported by our participants and compute the means to see on average how frequently (Never, Seldom, Frequently, Always) our participants encounter any of the communication problems. When our participants were receivers, the mean is 1.90, with standard deviation 0.63. When our participants were senders, the mean is 1.67, with standard deviation 0.86. On average, receivers encountered any of the communication problems more frequently than senders of information.



(a) Receivers of information (Q7).



(b) Senders of information (Q12).

Figure 5.15: 95% confidence interval for the (F + A) proportion of communication problems reported by our participants when they were: a) receivers of information and b) senders of information.

In addition, we compute the mean responses for each of the problems listed in Q7 and Q12 by the (N + S) and (F + A) categories. The values are 0 and 1 respectively. The means are 0.33 (manageability), 0.29 (understandability), 0.24 (competency), 0.48 (accessibility), 0.43 (accuracy), and 0.48 (relevance). The overall mean for the problems listed in Q7 is 0.37, with a standard deviation 0.31. The mean responses indicate that the participants encountered problems more than "seldom" when they were receivers of information. The most common problems are accessibility and relevance. The means for the problems listed in Q12 are 0.14 (manageability), 0.48 (understandability), 0.10 (competency), 0.33 (accessibility), 0.29 (accuracy), and 0.10 (relevance). The overall mean is 0.24, with a standard deviation 0.28. The most common problem encountered by the participants is understandability, in which the mean is much higher than other mean values in Q12 as well as the mean value for understandability in Q7. However, the mean responses for other problems in Q12 are lower than their counterparts in Q7. The means suggest that the participants have less difficulty when they were senders of information.

The analysis suggests that there might be a difference in the occurrence of communication problems in a software project depending on the sender or receiver role of our participants. A statistical analysis of the difference can be found in our paper on the survey [114] which reveals an apparent asymmetry in responses to Q7 and Q12. In the paper, we assume the Likert responses of never, seldom, frequently, and always are probabilities of occurrence with ratios 0:1:2:3. The overall mean for the problems listed in Q7 is 1.32, with a standard deviation 0.50. The overall mean for the problems listed in Q12 is 1.10, with a standard deviation 0.50. A paired-samples t-test value is 0.002, which indicates that the difference in the occurrence of communication problems is statistically significant with high confidence [114].

In summary, the analysis gives mild support to our hypothesis that a majority of stakeholders frequently or always encounter communication problems related to transparency problems. The most common communication problems reported by our participants were problems with accessibility, understandability, and relevance. The confidence intervals and mean responses indicate that our participants encountered any transparency problems more often than seldom. The statistical test supports our hypothesis that a majority of our participants frequently or always encounter some transparency problems.

5.3.5 Familiarity of Transparency in Different Contexts

We consider Q15 of the questionnaire to test SH2, whether a majority of stakeholders are familiar with transparency used in more than one context. We asked participants in Q15, "are you familiar with the term 'transparency' used in the context of...?" Figure 5.11

shows different contexts in which transparency appeared and the number of participants who were familiar with transparency in those contexts. To perform statistical analysis of SH2 using data from Q15, we divide the responses into two groups:

- **Group 1:** Participants unfamiliar with transparency or participants familiar with transparency used in only one context, and

- **Group 2:** Participants familiar with transparency used in more than one context.

We perform a frequency count to determine the number of participants in Group 1 (n_1) and the number of participants in Group 2 (n_2). Group 1 consists of participants who selected "I am not familiar with the term 'transparency' used in any context" or one of the transparency definitions listed in Q15. Group 2 consists of participants who selected more than one of the transparency definitions listed in Q15. Of our 21 participants, six were in Group 1 ($n_1 = 6$) and fifteen were in Group 2 ($n_2 = 15$). If our participants were representative of the population of software project stakeholders, we could conclude that about one-third ($n_1/21 = 0.29$) are familiar with at most one definition of transparency, and two-thirds know multiple definitions of this term. We also find that only two participants from Group 1 were not familiar with transparency. This suggests that almost all (19/21) of our participants were familiar with at least one definition of transparency.

To test SH2, we construct a null hypothesis. We hypothesise that there is no difference between Group 1 and Group 2. A binomial test reveals that the probability of finding 6 or fewer participants in Group 1 is 0.039, which is less than the 0.05 level of significance. The test gives us high confidence to support SH2, in which the proportion of Group 1 is less than 0.5 and the proportion of Group 2 is at least 0.5.

In addition, we refine SH2 to see whether our participants in Group 2 were familiar with different definitions of transparency. All 15 participants reported that they were familiar with transparency in the context of government, business, and ethics; in those contexts transparency has the meaning of making information visible. Of these 15 participants, 11 participants were also familiar with transparency used in computing which has the paradoxical meaning of hiding information from people.

In summary, the statistical analysis supports our hypothesis SH2. At least 50% of the participants were familiar with transparency used in more than one context. About 70% (15/21) of our participants were familiar with transparency in the context of government. Among participants in Group 2, about 70% of our participants (11/15) were also familiar with the definition of transparency in the context of computing.

5.4 Threats to Validity

In this section we discuss threats to the validity of the survey and mitigations to these threats. We consider the four main types of validity as discussed by Wohlin et al. [122] in the following subsections.

5.4.1 Conclusion Validity

According to Wohlin et al. [122], conclusion validity is concerned with the statistical relationship between the treatment and the outcome of an experiment. In the survey, we need to be aware of any threats that could affect the conclusions made about our hypotheses, SH1 and SH2.

The main threat to conclusion validity is the conversion of ordinal scale (e.g. Never, Seldom, Frequently, Always) to numerical values. We perform statistical tests using the transformed numerical values, which could violate statistical rules for analysing ordinal data. This could affect the results of the statistical analysis. In the analysis of our survey, we transform the Likert scale for Q7 and Q12 into numerical values for testing SH1 (whether a majority of stakeholders frequently or always encounter transparency problems). We convert the Likert scale into dichotomous variables: 0 for "Never or Seldom" and 1 for "Frequently or Always". We also use a binomial test for any differences between the two proportions. Similarly, we use a binomial test for testing SH2. According to Kitchenham and Pfleeger [59], converting an ordinal scale to dichotomous variables is one approach to avoid scale violations. Therefore, the conversion of ordinal scale to numerical values is reasonable for analysing our data.

5.4.2 Internal Validity

Internal validity is concerned with whether there is a causal relationship between the treatment and the outcome of an experiment [122]. We need to ensure that the outcome is derived from the controlled variables of the experiment and not from the results of uncontrolled or unknown factors.

However, it is difficult in the survey to determine any causal relationship. This is because we did not have control over the independent variables. Moreover, we did not construct any causal hypotheses in the survey. We could only observe the differences in the variables. Therefore, there are no threats to internal validity in the survey.

Determining causation of problems in communication is future experimentation and research. An experiment could also confirm that transparency actually affects the frequency or severity of communication problems in software engineering.

5.4.3 Construct Validity

Construct validity is concerned with how well the treatment and the outcome reflect the concept or theory behind an experiment [122]. In the survey we need to assess whether the questions constructed actually test the two hypotheses, SH1 and SH2.

The main threat to construct validity is the survey design, in which we rely on participants' observations in software projects. For example, Q7 and Q12 in the questionnaire are created to explore different types of communication problems from participants' personal experience. Similarly, in the demographics section, we ask participants to self-assess their knowledge and experience in software projects.

Another threat is the wording of the questions in the survey, which could also affect the conclusion validity of the survey. If the wording of the questions is ambiguous, participants could respond poorly to the questions. One possible threat to the validity of our analysis is the wording of Q7 and Q12. We intend to use Q7 and Q12 to explore participants' experience with the most recent software project in which they were involved. However, the wording of Q7 and Q12 does not make this clear. Some participants might have reported their experience with different software projects. The responses to Q7 and Q12 do not reveal any significant impact from this threat.

Although there was a possible confusion about "project", Q7 and Q12 should exhibit participants' perception of problems in communication. The two questions should also reveal whether there is a correlation between transparency and communication problems. The responses to Q7 and Q12 give us some evidence to test SH1 concerning the types of communication problems that participants have encountered in different software projects.

5.4.4 External Validity

External validity is concerned with generalising the results obtained from the experiment [122]. The main threat to external validity is the choice of using convenience sampling in the survey. We cannot generalise about the entire population involved in the software industry. This is because our participants are limited to one type of the target population, mainly software developers. Moreover, our participants might not be representative for the whole population of software developers due to small sample size. For example, the survey responses suggest that about two-thirds of our participants were familiar with more than one definition of transparency; the probability of finding less than half of our sample population who were not familiar with transparency is unlikely. We can generalise this finding to conclude that more than half of all software developers are likely to be familiar with more than one definition of transparency. However, this conclusion is subject to a threat of external validity due to a small sample size. Results that show less than half of other samples of software developers who are familiar with transparency could refute this conclusion.

Our results are limited to software developers. Our results might change if the type of our sample change. For example, if our participants are not software developers, but are all project managers, the responses to the the types of communication problems encountered by them (Q7 and Q12) might be different. We might find that project managers encounter manageability problems more frequently than other problems. They might also have other communication problems such as time differences and language barriers when their developers and clients are in disparate locations.

To address this threat, we can reuse our survey in the future with other convenience samples of software developers. We can also reuse our survey with other types of stakeholders such as end users and project managers in the future.

5.5 Summary

In this chapter we present our design of the survey, the execution of the survey, and statistical analysis of its results. The survey is conducted in the early stage of our research to explore communication problems that stakeholders have encountered in software projects. It is also used to evaluate our preliminary definition of transparency and to define a more precise scope for transparency in software engineering. In Section 5.3, we present the descriptive statistics of the responses as well as the hypothesis testing for SH1 and SH2.

The findings from the survey are indicative because of the small number of software project stakeholders who responded to the survey. The analysis suggests mild support to our hypothesis SH1, whether a majority of stakeholders frequently or always encounter transparency problems. The responses from the participants show that transparency is a common problem in communication. Problems with accessibility, understandability and relevance are the top communication problems reported by our participants. In addition to the transparency problems, our participants reported accuracy as another common type of communication problem. The mean responses for these problems are between 0.33 and 0.48, suggesting that these problems occur more than seldom in communication. Furthermore, the responses show that more than 50% of the participants were familiar with transparency used in more than one context. The results of this lend support our hypothesis SH2, whether a majority of stakeholders are familiar with transparency used in more than one context. We also find that many participants were familiar with two different meanings of transparency used in government and computing.

Despite the small sample size, the responses help us to gain insights into different types of communication problems. We also discover some interesting points for future investigation from the analysis of the responses. Furthermore, the findings from the survey enable us to improve our definition of transparency.

In the next chapter, we present an experiment for the evaluation stage of transparency. This experiment compares the effectiveness of two types of requirements documents with different degrees of transparency in presenting functional requirements of a software system. We present our design, execution and results of the experiment in the next chapter. We also discuss threats to the experiment's validity and summarise the findings.

6 An Experiment to Evaluate Transparency

In previous chapters we explored how transparency was defined in software engineering literature. We also conducted a survey to collect opinions about our preliminary definition of transparency. In this chapter we present our experiment to test whether transparency is an important attribute in the context of requirements engineering. The evidence collected from the experiment helps us to test our hypotheses about the importance of transparency in requirements engineering.

As discussed in Chapter 4.2.2, we design an experiment to compare different requirements documents. Two different types of requirements documents, requirements written in natural language and the use case model, present functional requirements of a software system to stakeholders. We hypothesise that the use case model is more effective than requirements written in natural language for stakeholders to answer questions about the functional requirements of a software system. This is because our preliminary transparency analysis in Section 6.1.5 suggests that the use case model is more transparent than requirements written in natural language.

This chapter has four main sections based on the reporting guidelines by Jedlitschka et al. [57] as with Chapter 5. The first section has a description of the experimental design. In this section, we also discuss our preliminary transparency analysis of the two types of requirements documents. In the second section we describe the execution of our experiment. We then present our analysis and findings in Section 6.3. Lastly, we discuss threats to validity of our experiment and summarise our findings.

6.1 Experimental Design

This section presents the planning of our experiment. The main goal and research questions are presented in Section 6.1.1. Section 6.1.2 describes the target population and the sampling method for our experiment. The experimental materials and the tasks to be performed by participants are presented in Section 6.1.3 and Section 6.1.4. The hypotheses to be tested in our experiment are discussed in Section 6.1.5. In Section 6.1.5, our preliminary analysis of which requirements document is more transparent is also discussed. In the last part of this section, the type of experimental design as well as ethical considerations are discussed.

6.1.1 Goal

The experiment is designed to collect evidence to support or refute one of the hypotheses derived from RQ3, how important is the concept of transparency to successful software development? The main hypothesis to test in the experiment is:

> **A transparent requirements document is more effective for developers to answer questions about the requirements of a software system than a non-transparent requirements document.**

We aim to compare different types of requirements documents which have different degrees of transparency in presenting software requirements to developers. We further refine the hypothesis into the following questions:

> **EQ1.** Will participants spend less time in answering questions about requirements of a software system using a more transparent requirements document?

With this question we compare the time spent by participants to answer questions in the experiment. We aim to use time measure as an indication of how well the requirements documents help participants to answer their questions.

> **EQ2.** Will participants answer questions more correctly using a more transparent requirements document?

We construct this question to compare answers made by participants using different types of requirements documents. This question aims to compare the effectiveness of requirements documents in helping participants to answer questions correctly.

> **EQ3.** Will participants be more confident about their answers using a more transparent requirements document?

With this question we test whether participants who use a more transparent requirements document are more confident in their answers. We plan to answer this question by comparing the responses made by participants on how well they think they answered the questions correctly.

6.1.2 Participants

This subsection presents the target population for the experiment. The sampling method for the experiment is also described.

Target Population

The target population for the experiment is software professionals. For example, software developers, requirements engineers, and test analysts are our potential participants. We aim to recruit participants from software industry because they are likely to have experience in using different types of software artefacts for their work.

In addition to software professionals, we aim to recruit tertiary students in software engineering, computer science, information technology, or other related areas. Students studying in these areas are likely to be familiar with software artefacts or parts of software artefacts. This is because they are often required to read requirements, software models, or code for their assignments and projects. Training of students is therefore not required for our experiment. Moreover, it is easier to recruit a large number of students within the time and financial constraints of the research.

Sampling Method

Similar to the sampling method of our survey, convenience sampling is used for the experiment. We consider any software professional who is willing to participate as a potential participant. Any tertiary student who is majoring in software engineering-related degrees and is willing to participate is also considered a potential participant.

6.1.3 Experimental Materials

The experiment involves the use of two different types of requirements documents and a questionnaire. The first requirements document is an actual requirements document (UAM IMS Requirements Specification) which describes an integration of an accommodation management system (UAM) and an identity management system (IMS) for a particular organisation. The second document is a use case model, which we created using the information from the requirements specification document. A questionnaire is also constructed for participants to answer questions about the requirements documents. In the following sections, we describe these materials for the experiment in detail.

UAM IMS Requirements Specification

The UAM IMS Requirements Specification (ReqSpec) document is written in natural language (see Appendix F for the complete document). It does not follow any specific formats or standards. The document is 39 pages long and contains four main sections:

1. Summary

   The first section is a summary of the integrated system. This section contains information about the background, scope, dependencies, references, and a glossary of terms used.

2. Solution Overview

   This section provides a description of the integrated system. It describes how the identity management system is used in the accommodation management system.

3. Functional Specification

   In this section, information about the functional requirements of the integrated system is presented. This section is the major part of the document, which contains the process models, business rules, assumptions, functional requirements, and data requirements for the system. It also includes screenshots of the user interface as well as non-functional requirements and test scenarios.

4. Approval and Change Control

   The last section of the document is a record of the changes made in the document. The version number of the document, description of change and the authors are recorded in this section.

The organisation that created the ReqSpec document and implemented the integrated system provided the ReqSpec document for use in the experiment. To protect the anonymity of the organisation, we first read through the original document and highlighted parts that identified the organisation. We then changed the highlighted parts including the name of the organisation and names of systems specific to the organisation. We also modified the screenshots of the integrated system to remove the logo of the organisation as well as any text that might identify the organisation. Finally, we removed the names of people who were involved in producing and reviewing the document.

Use Case Model

The use case model (UCM) document is created by extracting information from the ReqSpec document (see Appendix G for the complete document). We chose the use case model as the second requirements document type because it is used to capture functional requirements of a software system [45].

To construct use cases for our experiment, we follow the template guidelines by Anda et al. [5]. The template guidelines include a template for describing an actor and a template for describing a use case.

The UCM document has 17 pages in total. It contains five main sections:

1. Summary

   The first section is a summary of the integrated system. This section provides information about the background of the accommodation management system and the identity management system. It also provides a glossary of terms used in the document. The information presented in this section is the same as the information presented in the ReqSpec document.

2. Solution Overview

   In this section, an overview of the integrated system is presented. This section has the same information as the solution overview section provided in the ReqSpec document.

3. Use Case Diagram

   This section presents a use case diagram of the integrated system. The use case diagram is constructed using the UML Use Case template in Microsoft Visio. The diagram illustrates five actors and 11 use cases that we identified from the ReqSpec document.

4. Actors

   The fourth section of this document provides information about the actors involved in the use cases. The template guidelines provided by Anda et al. [5] are used for describing the actors. Each actor involved in the integrated system has a brief description and an example. Information about the actors is extracted from the ReqSpec document with minimal changes to the original text.

5. Use Cases

   In the last section of this document, 11 use cases that are constructed using the template guidelines [5] are presented. Each use case is presented in a table form with information about actors, trigger, prerequisites, post-conditions and normal flow of events. Any variations and associations with the use case are also included in the table. All use cases are based on the original text of the ReqSpec document.

Questionnaire

The questionnaire for our experiment contains 23 questions for participants from software industry and 26 questions in total for student participants. Some of the questions are optional if participants run out of time (see Appendix E for the complete questionnaire). To minimise bias in favour of the UCM document, the questionnaire is constructed based on the wording in the ReqSpec document (see Section 6.4.3 Construct Validity for more detail on how we minimise such bias). The questionnaire is divided into the following sections:

1. Demographics

   There is a different version of the first section, one for participants from the software industry and another for students. For participants from the software industry, there are six demographic questions in total. Participants are asked to answer questions about aspects of a software project that they are involved in, their roles in a software project, and their years of working in the software industry. They are also asked about how they get to know software requirements and how effective different types of documents or models are in helping them to understand functional requirements. The last question of this section is optional for participants to answer. The question asks for types of documents or models that participants prefer to use for understanding functional requirements of a software product.

   For student participants, there are nine questions in this section. Student participants are asked questions about their tertiary institution, degree and major, and year of study. They are also asked to select from a list of software models or modelling methods that they have studied. In addition, we ask student participants for their work experience in the software industry. If student participants have worked or are currently working in the software industry, they are asked questions relating to their work experience. These questions are the same as the ones given to participants from software industry except for the first two questions about their involvement in a software project.

2. Part 1. Reviewing Functionality of a Software System (P1Q1 – P1Q8)

   This section is the main part of our experiment, where we set a 40-minute time limit on participants in answering questions. The purpose of this section is to help us to compare the effectiveness of the two requirements documents in terms of time (EQ1) and correctness (EQ2). In this section we ask each participant to review one type of requirements document. Participants are asked to answer questions based on the information provided in the document. They are also asked to write down problems if they could not answer the question rather than leave it blank. In addition, we ask participants not to spend more than 10 minutes on question P1Q4.

   This section contains eight questions in total. The first question asks participants to write down the type of requirements documents that they receive at the start of the experimental session. Participants are then asked to record the time they start answering this section. From questions P1Q3 to P1Q7, participants are asked about the software system described in the requirements documents. The questions are organised in the order of ease in locating answers in the ReqSpec document. All questions, except P1Q7, have specific answers found in the ReqSpec document and the UCM document. There is no clear information from the ReqSpec document to answer P1Q7. The last question P1Q8 asks participants to record the time when they finish answering this section of the questionnaire.

3. Part 2. Overview of the Software Document (P2Q1 – P2Q9)

   In this section of the questionnaire, we ask participants nine questions about their opinions on the requirements documents. Only questions P2Q4 to P2Q6 are compulsory for participants to answer. The first three questions (P2Q1 – P2Q3) ask participants if the documents contain any duplicated or redundant information, if there are any inconsistencies or errors in the documents, and if any information is missing from the documents.

   We ask participants questions relating to the three attributes of transparency in P2Q4 to P2Q6. The three attributes, accessibility, understandability, and relevance, are not explicitly stated in the questions. In P2Q4, we ask participants if they have to go through different parts of the requirements documents to answer one of the questions from Part 1 (P1Q6). In P2Q5, we ask participants to rate how well the documents help them to identify information, to read only the relevant information that they need, and to understand the functionality of the software system. To answer our research question about participants' confidence (EQ3), we ask participants in P2Q6 to assess how well they think they answer questions in Part 1 correctly.

The last three questions (P2Q7 – P2Q9) of this section are optional. In P2Q7, participants can comment on any problems that they encounter if they ran out of time to answer questions in Part 1. In P2Q8, participants can comment on how much they like the requirements documents and how they would improve such documents. Finally, participants can also comment on any problems or concerns regarding software artefacts or communication with other stakeholders in P2Q9.

6.1.4 Tasks

The experiment involves participants reading either a ReqSpec document or a UCM document and answering a questionnaire. From our pre-test of the experiment, we estimate that the experiment takes up to one hour. The participants' main tasks are to answer the questionnaire and to read the requirements documents given to them at the beginning of the experimental session. Participants are not required to read everything provided in the documents. They need to read only the parts that they think can help them to answer questions in Part 1 of the questionnaire.

6.1.5 Experimental Hypotheses

This subsection describes the hypotheses to be tested in the experiment. The variables of the experiment are also described in this section. Before presenting the experimental hypotheses, we first discuss our preliminary transparency analysis of the two requirements documents for the experiment. The purpose of the preliminary transparency analysis is to help us determine which requirements document is more transparent for the experimental hypotheses.

Preliminary Transparency Analysis

The main goal of the experiment is to test whether a more transparent document allows stakeholders to answer questions more effectively than a less transparent document. To test this, we need to first determine if the ReqSpec document or the UCM document is more transparent. We determine which document is more transparent by comparing the documents in terms of each attribute of transparency:

- **Accessibility.**

  We evaluate the accessibility of requirements documents based on the three questions for determining the accessibility of information as discussed in Chapter 3. The first question relates to whether the communication channel is available to stakeholders. In our experiment, the ReqSpec document and the UCM document are both available to our participants.

  The second question is about the format in which the information is presented. We assume that our participants can use the documents which are in either paper or electronic format (PDF).

  The last question to determine the accessibility of information is stakeholders' ease in reaching the location in the information source within a reasonable amount of time. In the UCM document, the structure for describing actors and use cases is based on the template guidelines by Anda et al. [5]. We use these template guidelines because the results from the experiment conducted by Anda et al. suggest that it is easier for developers to find information in use cases that are based on the template guidelines than use cases that are constructed using other types of guidelines.

  We believe that our participants can more easily locate the page where they think the answer is by using the UCM document than by using the ReqSpec document. This is because the UCM document contains fewer pages than the ReqSpec document. Moreover, information about the functionality of the software system is presented in tables in the UCM document whereas information in the ReqSpec document is mostly presented in plain text. We believe that it is easier for participants to find a particular piece of information in the UCM document as participants can identify different parts of the information presented in tables better than the information presented in plain text.

  The UCM document seems to allow participants to locate information that is likely to answer their questions more easily than the ReqSpec document. Thus, we believe that the UCM document is more accessible than the ReqSpec document for stakeholders to obtain information to answer questions about the functionality of a software system.

- **Understandability.**

  The understandability of information is determined by stakeholders' ease in recognising the meaning of information within a reasonable amount of time. We believe that there is no difference for our participants in understanding information about a software system using the ReqSpec document and the UCM document. Although some participants might be unfamiliar with use case models, information presented in the UCM document is written in natural language which is the same as the information presented in the ReqSpec document. The description of the functional requirements in both documents contains little technical detail. We believe that our participants can understand what is presented in the documents within the experiment session. Moreover, according to Leffingwell and Widrig [63], use cases provide "related, cohesive threads of behavior, or scenarios, that can be understood by both the user and the developer". This suggests that any stakeholder can understand the information provided in use cases. Hence, we believe that the information presented in the ReqSpec document and the UCM document is understandable to our participants. There should be no difference in the understandability of information.

- **Relevance.**

  The relevance of information is determined by three questions. The first question is how quickly our participants can answer their questions. This question is difficult to answer because we need to measure and compare the time participants need to answer questions. We believe that participants using the UCM document can answer questions more quickly than participants using the ReqSpec document. We reason that there is less information in the UCM document that is irrelevant to the functionality of the system than in the ReqSpec document. We believe that participants spend less time reading information from the UCM document than from the ReqSpec document.

  The second relevance question is how directly connected the information is with stakeholders' questions. In the experiment, we set questions for our participants to answer. The questions are related to the functional requirements of a software system. Since use case model is used to capture functional requirements [45], we believe that the information in the UCM document is more directly connected to our questions than information in the ReqSpec document.

  The third relevance question is, does the information answer stakeholders' questions sufficiently? This question is difficult to answer because it is difficult to determine the helpfulness of this information in answering the questions. Both documents have the same information about the functional requirements of a software system that we think is sufficient for our participants to answer most of the questions in Part 1 of the questionnaire. However, we think that participants are more likely to read less irrelevant information using the UCM document than the ReqSpec document. This is because information about the same functionality of a software system is grouped in the same use case in the UCM document, whereas the information in the ReqSpec document is scattered. We think that participants using the ReqSpec document are likely to go to another location within the document to find information to answer their questions than participants using the UCM document. Therefore, we think the information provided in the UCM document is more relevant to our participants than the information provided in the ReqSpec document.

Table 6.1 summarises our preliminary transparency analysis. Based on the questions, we determine that the information in the UCM document is more accessible and more relevant to our participants than the information in the ReqSpec document. Therefore, the UCM document is more transparent in helping stakeholders to answer questions about the functional requirements of a software system than the ReqSpec document.

Attributes of Transparency	Questions	Summary of Analysis
Accessibility	Is the communication channel available for stakeholders to find answers to their questions?	Both ReqSpec and UCM documents are available to our participants.
	How easily can stakeholders use the format in which the information is presented?	The formats of the documents, either paper or PDF, can be used by our participants.
	How easy can stakeholders access information from the channel that they believe is likely to answer their questions within a reasonable amount of time?	The UCM document contains fewer pages and the information is presented in tables. The UCM document is better in terms of helping stakeholders to reach the location within the channel than the ReqSpec document.
Understandability	Once stakeholders obtain the information, how easily can they recognise the meaning of the information within a reasonable amount of time?	The description of the functional requirements contains little detail in the ReqSpec document and the UCM document. Our participants can understand the information presented in both documents within the experiment session.
Relevance	How quickly can stakeholders answer their questions using the information?	The UCM document has more relevant information to the functionality of the system than the ReqSpec document has.
	How directly connected is the information with stakeholders' questions?	The UCM document only contains information about the functional requirements of the system.
	Does the information answer stakeholders' questions sufficiently?	Information about the same functionality is grouped in the same use case in the UCM document whereas information is scattered in the ReqSpec document. Participants are likely to go to another location within the ReqSpec document to find information to answer their questions.

Table 6.1: Summary of the preliminary transparency analysis.

Hypotheses

To test the main hypothesis of our experiment, we derive the following hypotheses. Each hypothesis corresponds with the research questions EQ1, EQ2, and EQ3, respectively.

> **EH1.** There is a difference in the time spent by participants using the UCM document and participants using the ReqSpec document to answer questions in Part 1 of the questionnaire.
>
> **EH2.** There is a difference in the number of questions answered correctly in Part 1 of the questionnaire by participants using the UCM document and by participants using the ReqSpec document.
>
> **EH3.** There is a difference between the confidence of participants using the UCM document and the confidence of participants using the ReqSpec document in their answers to Part 1 of the questionnaire.

Variables

In our experiment, the independent variable for EH1, EH2 and EH3 is the type of requirements documents that we give to our participants. The dependent variables are the time spent by participants to answer Part 1 of the questionnaire for EH1; the number of correct answers made by participants in Part 1 of the questionnaire for EH2; and the confidence level by participants about their answers for EH3.

We are also observing the differences in experience between two groups of participants in our experiment. The first group consists of undergraduate students. The second group consists of graduate students and participants from software industry. We divide the participants into these groups because we believe that graduate students and industry participants are likely to have more interactions with different types of software artefacts than undergraduate students. We want to know if graduate students and industry participants spend less time on answering questions in Part 1 than undergraduate students, and whether graduate students and industry participants answer more questions in Part 1 correctly than undergraduate students.

6.1.6 Design

This section describes the type of design for the experiment. This section also discusses the ethical issues that arise from the experiment.

Type of Experimental Design

Our experiment is a between-subjects design in which each participant is subject to only one treatment, treatment ReqSpec or treatment UCM. That is, each participant reads only one type of the requirements documents, either a ReqSpec document or a UCM document. Each participant is assigned to one treatment instead of two treatments for the experiment to avoid carryover effects, where participants could become more accomplished through practice and experience [101].

Participants in our experiment can choose to participate either in-person or on-line, to be discussed in later sections. For the in-person experiment, the researcher hands participants printed copies of the consent forms, the requirements documents and the questionnaire. The researcher is present at all times during the experiment session to answer any questions. For the on-line experiment, the questionnaire is self-administered. Participants receive the requirements documents in PDF format and a link to the web-based questionnaire via email. Participants complete the web-based questionnaire on their own.

Ethical Considerations

Similar to our survey design in Chapter 5, we consider the following ethical issues that helped to shape the experiment:

1. Anonymity

   There is some small likelihood that participants' handwriting is recognisable to the researcher in the in-person experiment. To minimise this risk, consent forms are distributed and collected separately from the questionnaire. Participants are asked to put the consent forms face down on the table so the researcher cannot see their handwriting. The consent forms are stored separately from the questionnaire. Thus, the researcher cannot associate the names with participants' handwriting. Handwriting from the consent forms is not compared with the handwriting from the questionnaire.

   Moreover, to protect participants' privacy, we do not ask for any identifying information in the questionnaire. We do not record any identifying information such as IP addresses in the responses collected. We remove all identifying information disclosed from the responses. Furthermore, we analyse and report the information provided by participants anonymously.

2. Confidentiality

   Similar to our survey, all data collected are not made available to the public. Access to all the data collected for the experiment is limited to the supervisors and the researcher. The information provided by participants is reported anonymously.

3. Rights to withdraw

   Similar to our survey, participants cannot withdraw data from our experiment due to the anonymous questionnaire. We cannot identify individual participants and their responses once they returned the questionnaire to the researcher or submitted the questionnaire on-line.

4. Informed consent

   Participants are not required to sign consent forms for the on-line experiment. However, we include a consent page at the beginning of the web-based questionnaire which is similar to the format of the survey. The purpose of the consent page is to inform participants of what is involved in the experiment.

5. Conflict of interest

   There is a small likelihood that our participants are students of the supervisors of this research. To avoid conflict of interest, participants are contacted by the researcher only. The experiment sessions are conducted by the researcher only. The supervisors are not involved in any of the experiment sessions. Student participants are made aware that their grades would not be affected by taking part in our experiment.

6.2 Execution

This section describes the execution of our experiment. We describe the preparation for conducting our experiment. We also give an overview of the experimental procedure.

6.2.1 Preparation

To conduct our experiment, we applied and received ethics approval from the UAHPEC in May 2012. The application can be found in Appendix D, which contains information about the ethical considerations as discussed in the previous section as well as the consent forms, participant information sheet (PIS) and advertisements used in the experiment.

To recruit student participants, we advertised our experiment using posters within the University of Auckland campus. The posters were posted on notice boards in Department of Computer Science and Department of Electrical and Computer Engineering. We also asked two lecturers from Computer Science and Information System departments to advertise our experiment during one of their classes. To recruit participants from the software industry, we asked several software professionals to forward email invitations for participation to potential participants. Participation in the experiment was voluntary. To encourage people to take part in our research, we offered movie vouchers as an inducement for the in-person experiment.

In both posters and email invitations, we provided the researcher's email address for potential participants to contact. When receiving emails expressing their interest for participation, we replied with available times for the experiment and attached a PIS. The PIS contained information about the procedure of the experiment, data storage, rights to withdraw from our research, anonymity of responses, and contact details. When we scheduled the time with potential participants, we sent details about the location where the in-person experiment would be held.

We also prepared an on-line version of the experiment for potential participants who could not come in person. We created a web-based questionnaire on SurveyMonkey and electronic versions of the requirements documents in PDF format. The link to the web-based questionnaire and a copy of the requirements documents (either the ReqSpec document or the UCM document) were sent to participants who wished to participate on-line. Participants were not required to sign consent forms for the on-line experiment because of the anonymous nature of the questionnaire.

6.2.2 Procedure

The experiment was conducted from June to September 2012. Participants could take part in person or on-line.

In-Person Experiment

The in-person experiment was conducted on a one-to-one basis in meeting rooms or labs within the Department of Computer Science. Participants could sit anywhere they liked in the room as long as the researcher could not see their handwriting. The experiment involved the use of a printed questionnaire and printed copies of the requirements documents.

Before the start of the experiment, each participant was given a PIS, a consent form, and an assurance letter from the Head of Department of Computer Science. The consent form ensured that participants understood the conditions for taking part in the experiment. Participants were asked to sign and return the consent forms if they agreed to participate. The assurance letter assured participants that their grades or relationship with the University of Auckland would not be affected. Each participant was also asked to draw a piece of paper with the number 1 or 2 from a paper box. The number represented the type of document that participants would receive for the experiment. Once participants signed and returned the consent forms, each participant was given a copy of the appropriate requirements document and a questionnaire.

Before participants started answering the questionnaire, they were asked to follow the instructions written on the questionnaire. Participants were also encouraged to ask any questions during the experiment session. The researcher reminded them that it was not necessary to read the entire document. They were not required to fill in questions marked "optional". During the experiment session, the researcher reminded participants about the time allotted for the questionnaire.

Participants were asked to return the questionnaire and the requirements documents when they finished. Each participant received a movie voucher for his or her time and effort in the experiment at the end of the session.

On-line Experiment

The on-line experiment involved the use of a web-based questionnaire and electronic versions of the requirements documents. The questions in the web-based questionnaire were the same as those in the in-person experiment. The requirements documents were randomly assigned to participants.

Similar to our survey design, consent forms were not required for the on-line experiment. A consent page was presented to the participants in which they selected "agree" or "disagree" for taking part in the research. The questionnaire would proceed when the participants chose "agree" from the consent page.

Participants were not required to answer questions that were optional. Similar to our survey described in Chapter 5, participants could progress through the questionnaire using the "Next" and "Prev" buttons. Participants were asked to click the "Submit" button for completing the questionnaire when they reached the last web page. Any responses made without clicking the "Submit" button were not used in the research.

6.2.3 Deviations

Most of our data collection was performed according to the procedure described in Section 6.2.2. There was one participant who had to leave in the middle of the in-person experiment. The participant was asked to note down the time that he stopped answering the questionnaire and to leave the questionnaire as well as the requirements document with the researcher. The researcher returned the questionnaire and the requirements document to the participant when he returned. The participant was then asked to note down the time that he resumed the questionnaire.

6.3 Analysis

By October 2012, we recruited 58 participants. Three people from the software industry took part in the on-line experiment and 55 people, including seven from the software industry, took part in the in-person experiment. Each treatment has 29 participants.

The responses were transcribed into spreadsheets by the researcher. We removed three responses by students to one demographic question from the data set. The demographic question concerned how participants got to know software requirements. The responses were removed because the question was aimed at participants with work experience in the software industry; these student participants had no work experience.

To perform statistical analysis, Likert scale responses were transformed into numerical values. For example, Likert items such as "Very poor, Poor, Satisfactory, Good, and Very Good" were transformed into numbers 1, 2, 3, 4, and 5 respectively. We used the transformed values in parametric statistical tests such as t-tests, which according to Norman [81], could be used for Likert data without "coming to the wrong conclusion".

We also calculated how long participants took to answer questions in Part 1 of the questionnaire from the start times and the finish times as reported by participants. Furthermore, we assessed participants' answers to questions in Part 1 against the researcher's answers. Participants' answers were classified into different groups based on the correctness of answers as well as the locations of answers reported by participants.

In addition, we labelled the comments made by participants in the questionnaire with codes. The codes were based on our definition of transparency as well as any interesting points that arose in the comments. We then identified themes from the codes and grouped the codes according to themes. Coding enabled us to identify any common patterns relating to transparency from the experiment.

In the following sections we present the statistical analysis of the data collected from the experiment. We organise the analysis based on the structure of the questionnaire. We also present the results for testing the three hypotheses, EH1, EH2 and EH3. Finally, we discuss the themes identified from the experiment in the last section.

6.3.1 Demographics

10 people from software industry and 48 students participated in the experiment. We present the demographics for our industry and student participants in the following sections.

Industry Participants

Of the 10 industry participants, four people have zero to four years of experience and six have five to nine years of experience working in the software industry. Most of our industry participants reported that they were involved in more than one aspect of a software project. They were involved in design, development, testing and maintenance aspects of a software project. Figure 6.1 shows the number of industry participants involved in different aspects of a software project.

All industry participants reported that they held the role as developers in a software project at the time of the experiment. Some participants were also architects or requirements engineers. Figure 6.2 illustrates the number of these participants in different roles. We surmise that our industry participants might be working in small development teams as they have multiple roles in a software project.

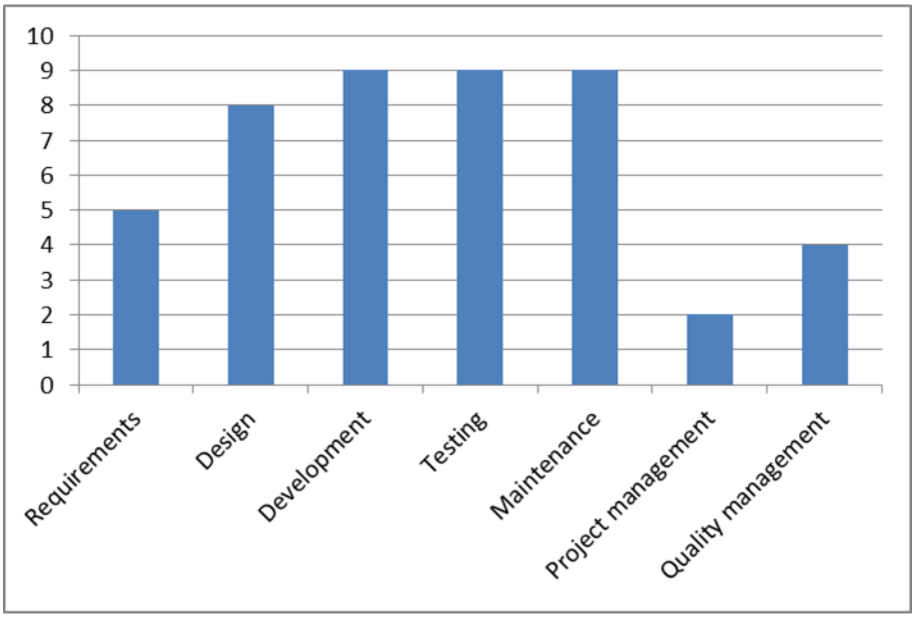


Figure 6.1: Number of industry participants involved in each aspect of a software project.

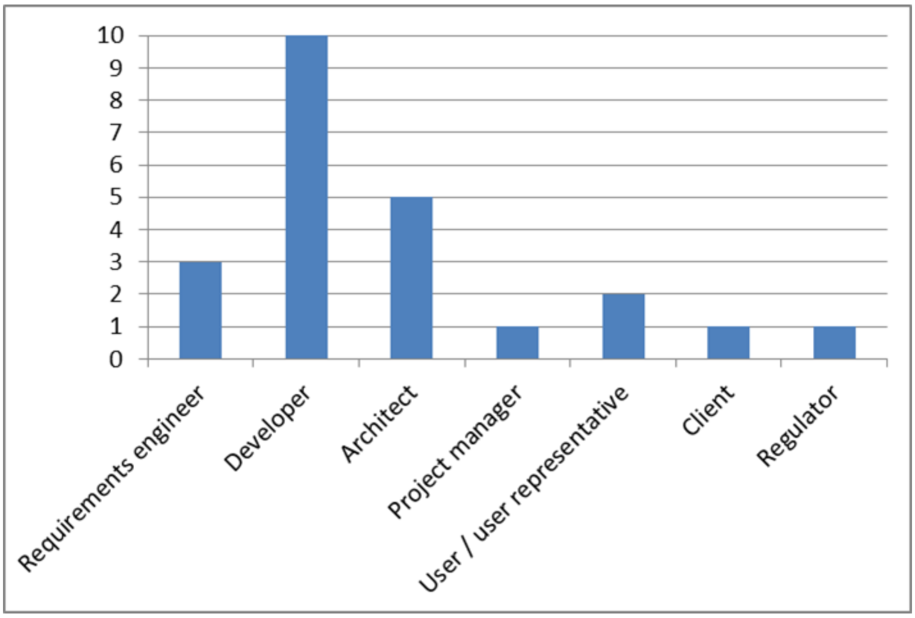


Figure 6.2: Number of industry participants involved in each role of a software project.

Student Participants

Most of our student participants came from the University of Auckland except for one who was an exchange student from France. Figure 6.3 shows the distribution of student participants by degree and major or specialisation. Of the 48 student participants, there were 21 graduate students and 27 undergraduate students. Out of the 21 graduate students, 14 were PhD students and 5 were Masters students. Of the 27 undergraduate students, most were specialising in Software Engineering. All of the undergraduate student participants were in their second year of study or above at the time of the experiment.

We asked student participants about what software models or modelling methods that they have studied. Most student participants reported that they learned more than one type of software model. More than 70% of students learned state machines, use case models and UML during their study. There were two other types of models that student participants reported. These were site-map and time automata. Figure 6.4 shows the percentage of student participants who learned each type of software models or modelling methods listed in the questionnaire.

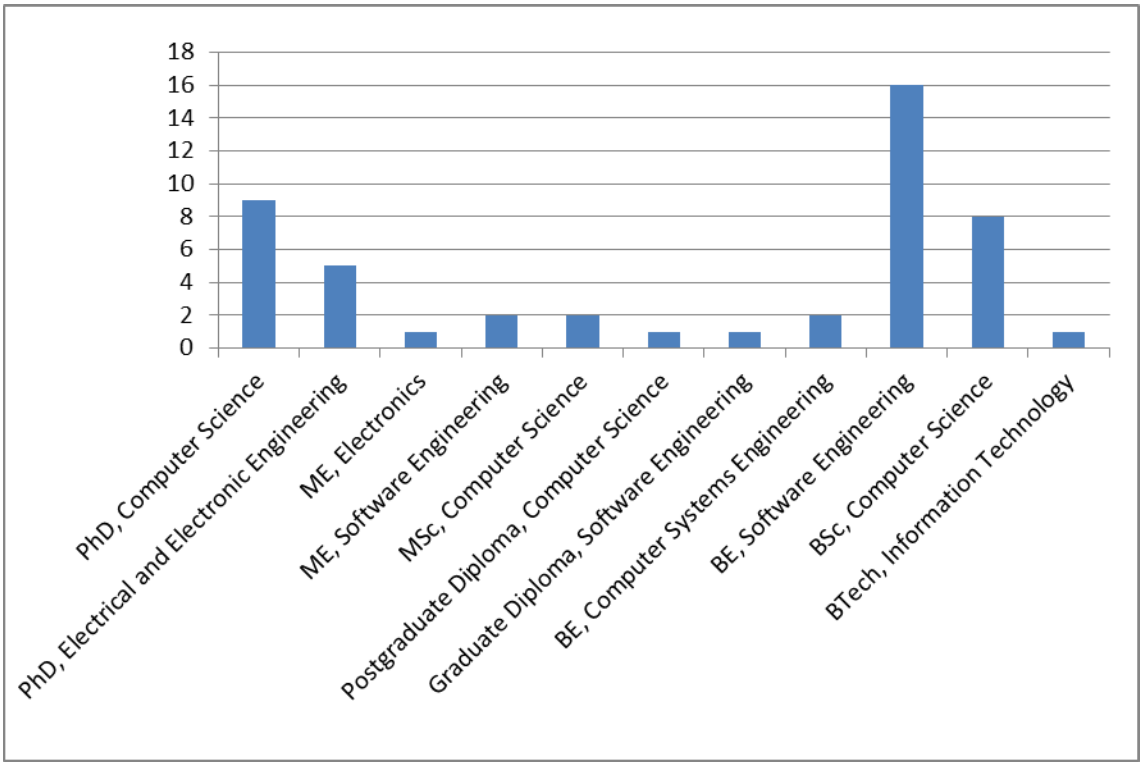


Figure 6.3: Distribution of student participants by degree and major or specialisation.

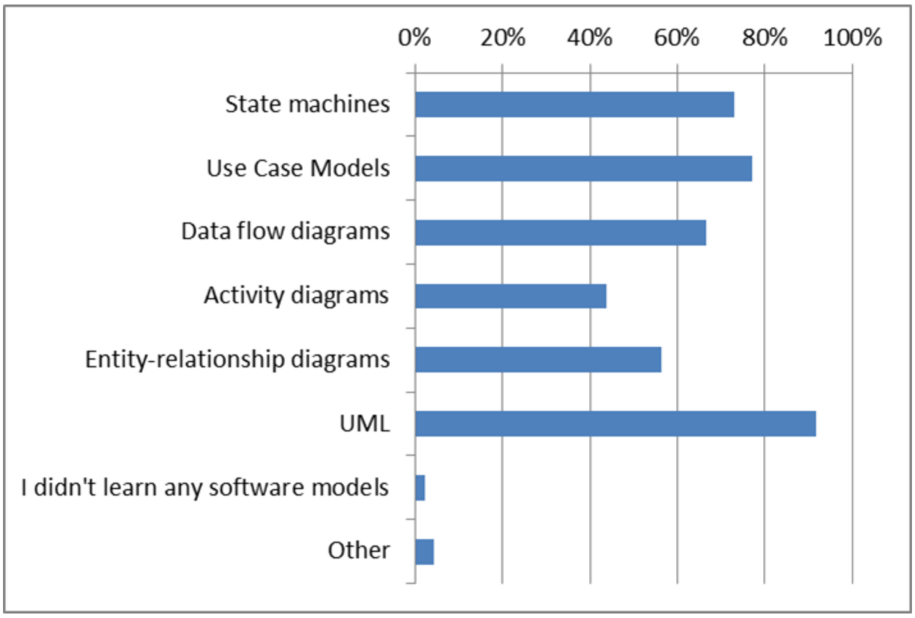


Figure 6.4: List of software models or modelling methods with respect to the percentage of student participants who learned such software models or methods during their study.

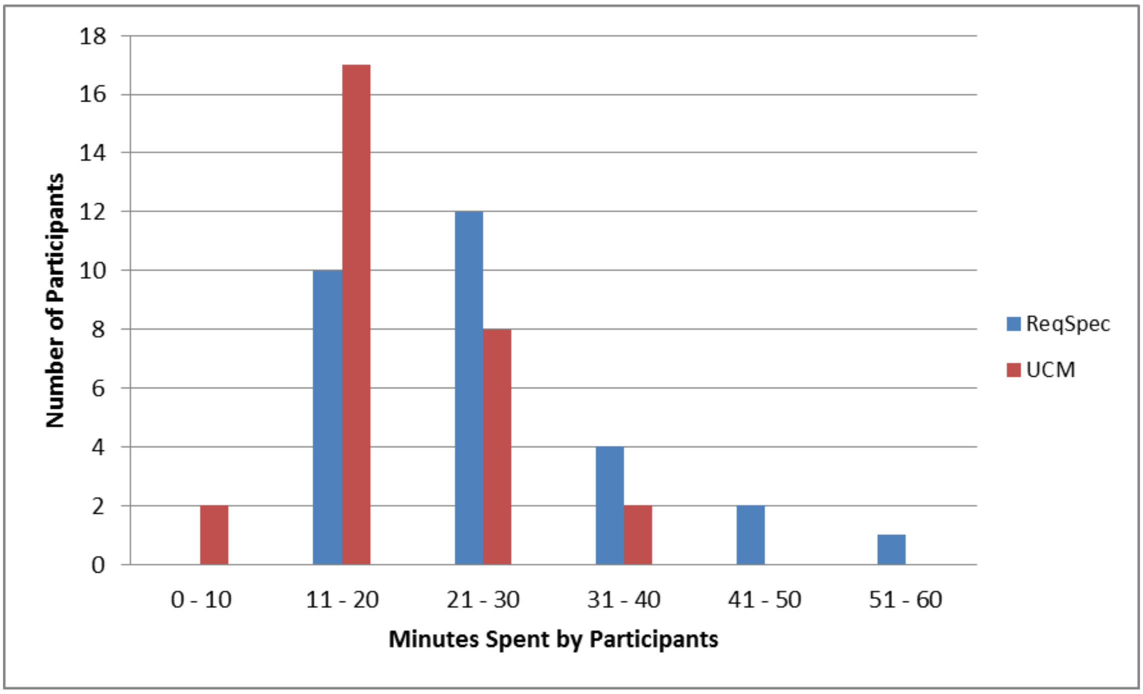


Figure 6.5: Distribution of participants using the ReqSpec document and the UCM document with respect to the time spent on Part 1 of the questionnaire.

6.3.2 Part 1. Reviewing Functionality of a Software System

In Part 1 of the questionnaire, we ask participants to read the requirements document and answer questions based on information in the document. In this section we present the results of Part 1 as well as the analysis for testing EH1, time spent by participants, and EH2, correctness of answers.

Time Spent by Participants

The time spent by each participant in answering Part 1 is calculated from the start time and the finish time recorded in questions P1Q1 and P1Q8. Figure 6.5 shows the distribution of participants using the ReqSpec document and the UCM document by the time spent on Part 1 in 10-minute intervals. As illustrated in Figure 6.5, more than 80% of the participants completed Part 1 within 30 minutes.

The mean time for treatment ReqSpec is 25.1 minutes, whereas the mean time for treatment UCM is 18.6 minutes. The standard deviations are 10.0 and 7.1 respectively. The longest time spent by participants to answer questions in Part 1 is 57 minutes. The shortest time spent by participants is 7 minutes. Table 6.2 summarises the descriptive statistics for the time spent in minutes by our participants.

The box plot in Figure 6.6 shows the distributions of the time spent by our participants.

Type of Document	Mean	Standard Deviation	Median	Min.	Max.
ReqSpec	25.1	10.0	23	12	57
UCM	18.6	7.1	17	7	40

Table 6.2: Descriptive statistics for the time spent in minutes by participants using the ReqSpec and the UCM documents to answer questions in Part 1 of the questionnaire.

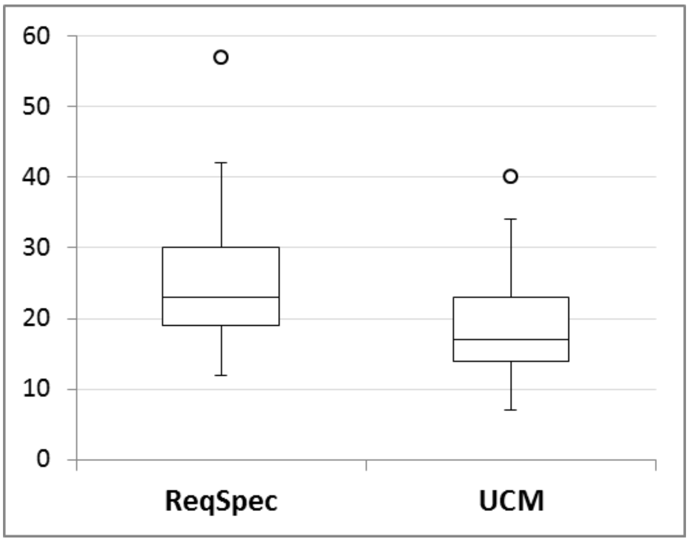


Figure 6.6: Distribution of the times spent by participants using the ReqSpec and the UCM documents to answer questions in Part 1.

We find that the time distributions for both treatments are concentrated on the lower end of the scale. The boxes overlap, but there appears to be a difference between the two treatments as the upper quartile of UCM has the same value as the ReqSpec median.

Figure 6.6 also shows an outlier for each treatment at the top of the scale. The longest time it took for participants in each treatment appears to be an outlier. However, we compute a new mean time for each treatment using SPSS which removes the top and bottom 5% of the time values recorded in the data set. The adjusted mean time for treatment ReqSpec is 24.3 and the adjusted mean value for treatment UCM is 18.05. When we compare the original mean values with the adjusted mean values, it seems that the outliers influence the means minimally. Therefore, we include the outliers as part of the data set for testing the hypotheses.

On average participants using the ReqSpec document spent more time on Part 1 than participants using the UCM document. In the following section, we present the statistical analysis for testing for a difference in the time spent by the ReqSpec documents participants and the UCM document participants.

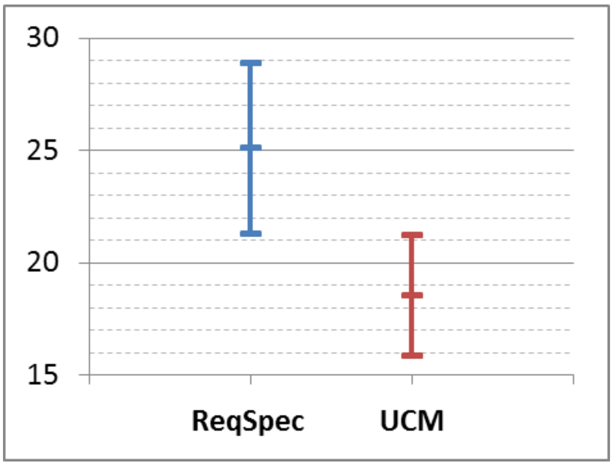


Figure 6.7: 95% confidence intervals for the mean time spent by participants using the ReqSpec document and participants using the UCM document.

Hypothesis Testing: Differences in Time (EH1)

We plot a 95% confidence interval graph as shown in Figure 6.7 to compare the means of the time spent by participants in each treatment. Figure 6.7 shows that the two confidence intervals do not overlap which suggests that there is a difference between the two treatments.

To test if the means are significantly different, we perform an independent-samples t-test. The null hypothesis for the test is: there is no difference between the mean time for using the ReqSpec document and the mean time for using the UCM document. We obtain a t-value of 2.88 with 56 degrees of freedom, and the two-tailed p-value is 0.006 which is significant at the 0.05 level. This indicates that there is a statistically significant difference between the two means. Therefore, we reject the null hypothesis that there is no difference in the means.

Examining the two means and the mean difference shows that participants using the ReqSpec document spent an average of 6.55 minutes more than participants using the UCM document. The 95% confidence interval for the mean difference indicates that we can be 95% confident that the actual difference in the time spent by participants using the ReqSpec document and participants using the UCM document is between 1.99 and 11.11 minutes. We also calculate the effect size using Cohen's d, which suggests a moderate to high practical significance ($d = 0.76$).

The statistical analysis supports our hypothesis EH1. The t-test gives us high confidence in a difference in the time spent between the two treatments. The mean values for the time spent by participants suggest that participants spent less time using the UCM document than the participants using the ReqSpec document.

Group	Number of Partici-pants	Mean	Standard Deviation	Median	Min.	Max.
Undergraduate	27	22.4	10.7	22	7	57
Graduate + Industry	31	21.3	7.8	20	12	42

Table 6.3: Descriptive statistics for the time spent in minutes by different groups of participants to answer questions in Part 1 of the questionnaire.

Time Spent by Experience

We wonder if there are differences in the time spent depending on the experience of our participants. We divide the participants into two groups: Undergraduate and Graduate + Industry. We hypothesise that there is a difference in the time spent by participants between Undergraduate and Graduate + Industry groups.

Table 6.3 shows a summary of the descriptive statistics for the time spent in minutes by different groups of participants. On average, Undergraduate participants spent more time than Graduate + Industry participants. The mean time for Undergraduate participants is 22.4 minutes with standard deviation 10.7. The mean time for Graduate + Industry participants is 21.3 minutes with standard deviation 7.8.

To see whether there is any difference in the time spent by different groups, we plot a 95% confidence interval graph as illustrated in Figure 6.8. Figure 6.8 illustrates that the two confidence intervals overlap, which suggests that there might not be any difference between the two groups. To test if there is any statistically significant difference between the two groups, we perform an independent-samples t-test. The null hypothesis for the test is that there is no significant difference between Undergraduate participants and Graduate + Industry participants in the time that they spent on answering Part 1 of the questionnaire. We obtain a t-value of 0.47, with 56 degrees of freedom, and the two-tailed p-value is 0.638. The mean difference is 1.15, and the 95% confidence interval of the difference is -3.73 and 6.04. Therefore, we fail to reject the null hypothesis. Further, Cohen's effect size value ($d = 0.12$) suggests low practical significance.

The statistical analysis suggests that there is no statistically significant difference in the time spent by the experience of our participants. The t-test fails to reject the null hypothesis, and the magnitude of the difference is small on the time-spent factor for the participants.

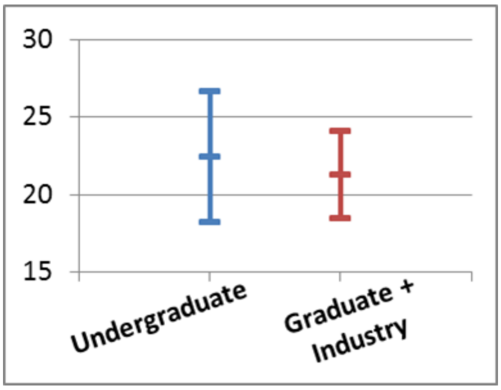


Figure 6.8: 95% confidence intervals for the mean time spent by Undergraduate participants and Graduate + Industry participants.

Type of Document	Correct	Incorrect	Total
ReqSpec	24	5	29
UCM	27	2	29
Total	51	7	58

Table 6.4: Number of participants who answered P1Q3 correctly using the ReqSpec document and the UCM document.

Responses to P1Q3

P1Q3 is the first question of the experiment that asks participants to find answers in the requirements documents. The question concerns the primary web authentication system for the organisation. We expect the answer to this question to be found directly in the *Background* section of *Summary* which is the first subsection of the first section for both documents.

Most of our participants in either treatments answered P1Q3 correctly. Less than 20% of our participants answered this question incorrectly. Table 6.4 shows the number of participants who answered P1Q3 correctly.

Responses to P1Q4

P1Q4 asks participants about the requirements for handling applications submitted in hard copies. Unlike P1Q3, P1Q4 requires participants to describe where and how they found the answer to this question. Participants are asked to note down the page numbers and section headings where they looked in the document.

The answer to P1Q4 is found in *3.4 Functional Requirements* under the *Functional Specification* section of the ReqSpec document. In the UCM document, the answer is found in *5.1 Make a new application* or *5.4 Manually enter an IMS ID* under the *Use*

Type of Document	Correct	Incorrect	Total
ReqSpec	20	9	29
UCM	26	3	29
Total	46	12	58

Table 6.5: Number of participants who found the correct location of the answer to P1Q4.

Cases section.

Since we did not specify how participants should report on the ways they obtained information, the responses varied in detail. For example, one participant using the ReqSpec document briefly described how he or she found the answer as follows:

> "1. Read table of contents.
>
> 2. Section 3.4 pg 8, 9.
>
> 3. Went through section 3.4 to list 10 for answer."

Similarly, one participant using the UCM document described only the section headings where he or she went through to find the answer. The participant has the following answer to P1Q4:

> "Content, solution overview, use case diagram (found corresponding use case), content, 5.4 Manually enter an IMS ID."

On the other hand, some participants described what information he or she was looking for in each page or section of the document. Here is such an example in this response by one participant using the ReqSpec document:

> "I first read through the table of contents for a section title that looked relevant (functional requirements) which I saw was in section 3.4 (pg 8). I then went to that page, and skimmed through the list of requirements, looking for anything relating to hard copies. The requirement is on pg 9, requirement #10."

To assess the correctness of answers to P1Q4, we looked at the locations of answers reported by participants. Answers that did not match our answer were recorded as incorrect. Table 6.5 shows the number of participants who found the location of the answer to P1Q4 correctly or incorrectly. For participants who used the ReqSpec document, 20 found the correct location. Nine participants who used the ReqSpec document recorded incorrect locations or reported that they have problems in finding relevant information to the question. For participants who used the UCM document, 26 found the correct location. Only 3 reported the locations incorrectly.

Type of Document	Correct	Incorrect	Total
ReqSpec	23	6	29
UCM	27	2	29
Total	50	8	58

Table 6.6: Number of participants who answered P1Q5 correctly.

Although more participants who used the UCM document found the correct location for P1Q4 than participants who used the ReqSpec document, there appears to be a misunderstanding about the word "requirements" in the question. Eight participants (5 Undergraduate participants and 3 Graduate + Industry participants) using the UCM document misunderstood "requirements" as the "prerequisites" of the use cases. However, all eight participants identified the location of information on handling hard-copy applications correctly.

Responses to P1Q5

The question P1Q5 asks participants to write down how hard-copy applications are being processed. The purpose of this question is to ensure that our participants did not guess the answer to P1Q4. It also helps us to see if our participants understand the question. The answer to this question is expected to be found in the same locations as the answer to P1Q4.

We evaluate the correctness of answers to P1Q5 against our answer. Table 6.6 shows the number of participants who answered P1Q5 correctly. For the ReqSpec document, 23 participants have the correct answer and 6 participants have an incorrect answer. Of the six participants, two participants reported that they could not find the answer in the ReqSpec document. For the UCM document, 27 participants answered the question correctly and two answered it incorrectly.

	ReqSpec	UCM	Total
No answer	4	0	4
Incorrect answer	11	4	15
Correct answer but wrong location	7	0	7
Correct location but wrong answer	2	8	10
Correct answer	5	17	22
Total	29	29	58

Table 6.7: Distribution of participants who answered P1Q6 by the correctness of their answers.

Responses to P1Q6

After participants find the requirements for handling hard-copy applications, in P1Q6 we ask them if NCEA exam results are displayed to applicants. We also ask them to note the page numbers as well as section headings where they found the answer.

Locating the answer to P1Q6 is harder than it is for P1Q3. The answer is found in *3.4.2 Completing an Accommodation Application* under the *Functional Specification* section of the ReqSpec document. In the UCM document, the answer is in the post-conditions of *5.9 Retrieve Secondary School details and NCEA results* under the *Use Cases* section. The answer to P1Q6 is "No".

We evaluate the correctness of participants' answers by looking at their responses to P1Q6 as well as the locations of their answers. As shown in Table 6.7, we classify participants' answers into five categories. For participants who used the ReqSpec document, four participants did not have answers or they were uncertain about the answer. Of the 29 participants who used the ReqSpec document, 11 reported incorrect answers with locations that did not match the actual answer. Seven participants using the ReqSpec document did not find the right location of the actual answer but answered the question correctly. This might be because participants inferred the answer from other information such as screenshots in the ReqSpec document. Also, two participants using the ReqSpec document did not answer the question correctly but reported the right location of the actual answer. Approximately 17% of the participants using the ReqSpec document answered the question correctly with the correct location.

The distribution of participants using the UCM document who answered P1Q6 correctly differs from the distribution of participants using the ReqSpec document. Of the 29 participants who used the UCM document, four participants have neither the correct answer nor the correct location. There are no participants who have the correct answer

	ReqSpec	UCM	Total
Clear answers about who can run reports	12	19	31
Can't find information	14	1	15
Information is not specified or not written	3	9	12
Total	29	29	58

Table 6.8: Number of participants who answered P1Q7 in three different types of responses.

but the wrong location of the actual answer. However, we find eight participants using the UCM document reported the correct location of the actual answer, but their answers were incorrect. Possibly participants did not read the information in the use case carefully. More than 50% of the participants who used the UCM document answered the question correctly with the correct location.

Responses to P1Q7

The last question that we ask of participants is, "who can run reports from the UAM system?" In this question we also ask participants to note the page numbers and section headings where they found the answer. This question might be tricky to participants because there is no clear information specified in either the ReqSpec document or the UCM document.

We classify the responses to P1Q7 into three types as shown in Table 6.8. The first type of responses belongs to participants who clearly answered the question about the actors for running reports. An example of a clear answer to P1Q7 by a participant using the ReqSpec document is as follows: "The University will be able to run reports...". 12 participants using the ReqSpec document and 19 participants using the UCM document gave clear answers. In particular, 14 participants who used the UCM document reported that no one could run reports, or the participants assumed the type of actors who could run reports. Example responses by participants using the UCM document include, "Not specified under 5.11 Run report nor in use case diagram. I would assume UAM admins"; "No one? ... use case not linked to any person(s)..."; "No one. Most likely auto-run"; "No one (empty actors field)...". The reason for these responses might be related to the presentation of information in the UCM document where the actor field in the *5.11 Run report* use case is empty.

The second type of responses concerns participants who could not find information to answer P1Q7 and did not explain why. Fourteen participants using the ReqSpec document could not find information, whereas only one participant using the UCM document could not find information about who could run reports. Example responses of the second type include, "... I cannot find answer for this question"; "Could not find information under 'Reports' on page 26".

The third type of responses is about the information not being specified or written in the documents. Participants used reasoning for these answers. Nine UCM document participants reported that the information was not specified or written in the document. One participant also commented that "the document may contain error as it has a section. But without any 'actor' ..." Other participants who used the UCM document made the following comments:

> "It is not specified clearly. The only mention is that 'the business' will be able to make use of the reports...",
>
> "Unknown. I could not find the answer for this as the actors section was blank on the use case on page 17. I also consulted the diagram but it also showed no connection to any actors."

On the other hand, of the 29 participants who used the ReqSpec document, three participants reported that the information was not specified or written in the document. One of these participants commented that "I can't actually find who is allowed to run reports. I guess it is inferred in various places that only staff members using the UAM can, but I cannot find any direct quote stating this..."

The reason for these differences might be related to the presentation of information in the UCM document, in which the actor field is explicitly stated. On the other hand, the ReqSpec document does not contain any specific fields about the actors.

Total Number of Correct Answers	**ReqSpec**	**UCM**	**Total**
0	1	0	1
1	6	0	6
2	5	5	10
3	12	9	21
4	5	15	20
Total	29	29	58

Table 6.9: Distribution of participants using the ReqSpec document and participants using the UCM document by the total number of correct answers from P1Q3 – P1Q6.

Hypothesis Testing: Differences in Correctness of Answers (EH2)

To test for any differences in the number of questions answered correctly by participants using different types of requirements document (EH2), we first calculate how many questions from P1Q3 – P1Q6 each participant answered correctly. We exclude P1Q7 because it is not possible to answer this question. Table 6.9 shows the distribution of participants by the total number of correct answers.

As shown in Table 6.9, of the 58 participants, 41 participants (approximately 70%) answered three or more questions correctly. For participants using the ReqSpec document, 17 participants have three or more correct answers, whereas for participants using the UCM document, 24 have three or more correct answers. This shows that participants using the UCM document answer more questions correctly than participants using the ReqSpec document.

We plot a 95% confidence interval graph as illustrated in Figure 6.9. We use the confidence interval graph to see if the means for the number of correct answers by participants using the two documents are different. Figure 6.9 shows that there is no overlap between the two confidence intervals. This suggests that there is a difference in the number of correct answers.

The mean value for the ReqSpec document is 2.48 with a standard deviation of 1.12, whereas the mean value for the UCM document is 3.34 with a standard deviation of 0.77. We perform an independent-samples t-test, where the null hypothesis is that there is no difference in the number of questions answered correctly by participants using different documents. The t-test shows that there is a significant difference between participants using the ReqSpec document and participants using the UCM document in the number of correct answers. The test gives us a t-value of -3.41 with 49.55 degrees of freedom. We obtain a two-tailed p-value of 0.001 which is less than the 0.05 level of significance. Therefore, we reject the null hypothesis. Furthermore, the mean difference is -0.86,

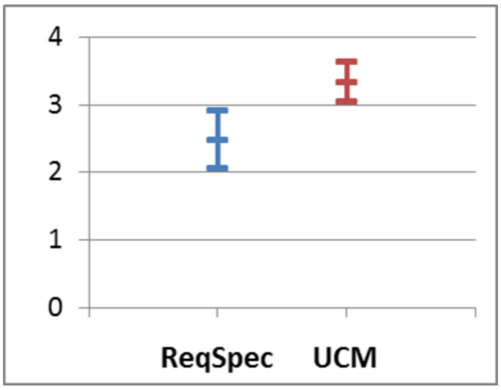


Figure 6.9: 95% confidence intervals for the number of correct answers by our participants in treatment ReqSpec and treatment UCM.

and the 95% confidence interval for the difference is −1.37 and −0.36. We can see that participants using the ReqSpec document answered an average of 0.86 questions less correctly than participants using the UCM document. Moreover, Cohen's effect size value is 0.89, which suggests high practical significance.

The analysis indicates a statistically significant difference in the number of questions answered correctly by participants using different documents. This in turn gives us high confidence to support our hypothesis EH2, correctness of answers. The magnitude of the difference is also significant. The mean values suggest that participants using the UCM document answered more questions correctly than participants using the ReqSpec document.

Correctness of Answers by Experience

We observe the difference in the number of correct answers by different groups of participants. Figure 6.10 illustrates how well each group of participants answers the questions from P1Q3 – P1Q6. The Graduate + Industry participants appear to answer more questions correctly than the Undergraduate participants. The mean value for the number of questions answered correctly by Undergraduate participants is 2.74 with a standard deviation of 1.06. The mean value for Graduate + Industry participants is 3.06 with a standard deviation of 1.03. It seems that on average participants with more experience answered more questions correctly than participants with less experience.

However, the 95% confidence interval graph as shown in Figure 6.11 suggests that there might not be any significant difference between participants with different level of experience as the two confidence intervals overlap. To test whether the difference is statistically significant, we perform an independent-samples t-test. The null hypothesis for this test is that there is no difference in the number of correct answers by different

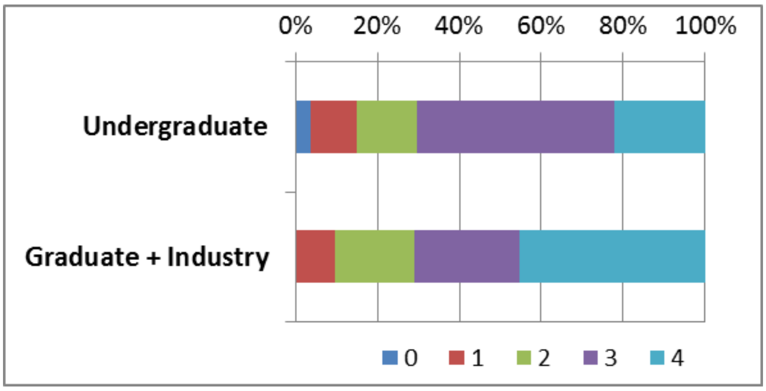


Figure 6.10: Total number of questions (P1Q3 – P1Q6) answered correctly by Undergraduate and Graduate + Industry participants.

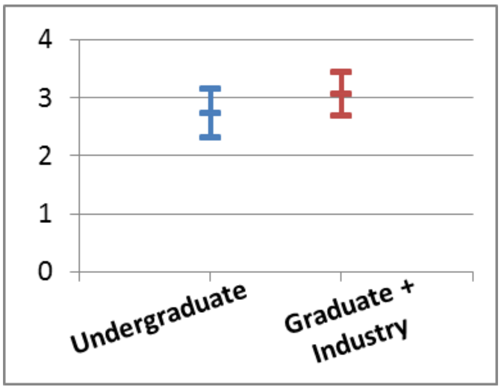


Figure 6.11: 95% confidence intervals for the number of correct answers by Undergraduate and Graduate + Industry participants.

groups of participants. We obtain a t-value of -1.18, with 56 degrees of freedom, and the two-tailed p-value of 0.244. This suggests that there is no significant difference at the 0.05 level of significance. Hence, we cannot reject the null hypothesis. The mean difference is -0.32, and the 95% confidence interval of the difference is -0.87 and 0.23. Further, Cohen's effect size value ($d = 0.31$) suggests low to moderate practical significance.

The statistical analysis shows that the number of correct answers by our participants is not affected by their experience. The statistical tests show that the difference is not statistically or practically significant.

6.3.3 Part 2. Overview of the Software Document

In Part 2 of the questionnaire, we ask participants to assess how helpful they think the requirements documents are to answer Part 1 questions. In this section we present the results of Part 2 based on the three attributes of transparency. We also present the analysis for testing EH3 (confidence of participants).

Accessibility of Information

As discussed previously in Section 6.1.5, we believe that the UCM document is better for our participants to locate the page where they think the answer is than the ReqSpec document. In question P2Q5a, we ask our participants how helpful the given document is to identify the information that they might need to answer questions in Part 1.

Figure 6.12 shows participants' assessment on how well the documents were in helping participants to identify information. Of the 29 participants using the ReqSpec document, 10 participants rated it good or very good whereas 21 participants using the UCM document rated it good or very good. Nine participants using the ReqSpec document reported it poor or very poor. For participants using the UCM document, one reported it poor or very poor. Some participants also commented on how the documents helped them to identify information. For example, one of the comments from participants using the UCM document is "Contents & Use case diagram helped to identify the sections".

We compute the mean values for the two treatments. The means are 3.03 and 3.90 with standard deviations of 0.94 and 0.77 for the ReqSpec document and the UCM document respectively. The mean values suggest that the ReqSpec document and the UCM document were more than satisfactory for our participants on average.

We also compare the two means by plotting a 95% confidence interval graph as shown in Figure 6.13. The confidence intervals show no overlap which suggests that there is a difference. To test whether the difference is statistically significant, we perform an independent-samples t-test. The null hypothesis for the test is that there is no difference in participants' assessments on the accessibility of information using the ReqSpec document and the UCM document. The t-test ($t = -3.81$, $df = 56$) indicates that the difference is statistically significant (0.000) at the 0.05 level of significance. The mean difference is -0.86, and the 95% confidence interval of the difference is -1.32 and -0.41. Further, Cohen's effect size value ($d = 1.01$) suggests a high practical significance.

The analysis shows that there is a difference in the accessibility of information using different requirements documents. Since the UCM document mean is greater than the ReqSpec document mean, the UCM document is better than the ReqSpec document in terms of helping participants to identify the desired information.

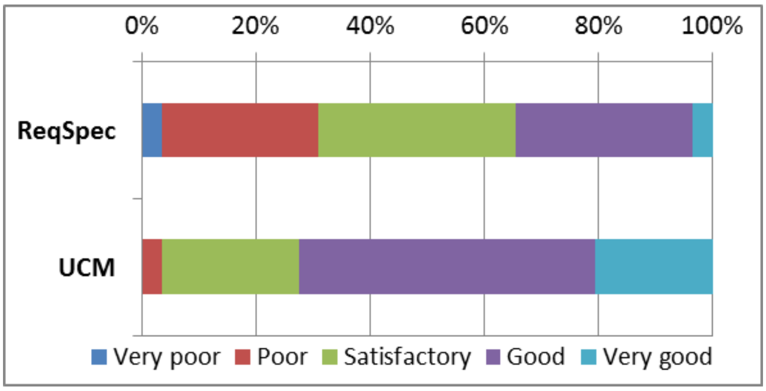


Figure 6.12: Participants' assessments on how well the ReqSpec document and the UCM document were in helping participants to identify the desired information to answer questions in Part 1 (P2Q5a).

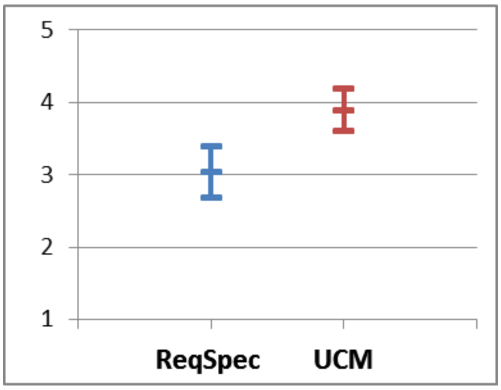


Figure 6.13: 95% confidence intervals for the responses to P2Q5a by participants using the ReqSpec document and the UCM document.

Accessibility of Information by Experience

We compare different groups of participants' assessments of the requirements documents in helping them to identify information as shown in Figure 6.14. Approximately 25% of our Undergraduate participants rated the documents poor whereas less than 10% of our Graduate + Industry participants rated the documents very poor or poor. More than half of our participants reported the documents as satisfactory or good.

The mean value for Undergraduate participants is 3.37 and the mean value for Graduate + Industry participants is 3.55 with standard deviations of 1.08 and 0.85 respectively. The mean values suggest that the documents were more than satisfactory for both groups of our participants on average.

A 95% confidence interval graph as illustrated in Figure 6.15 compares the means. The confidence intervals in Figure 6.15 overlap which suggests that there may be no significant difference between the means.

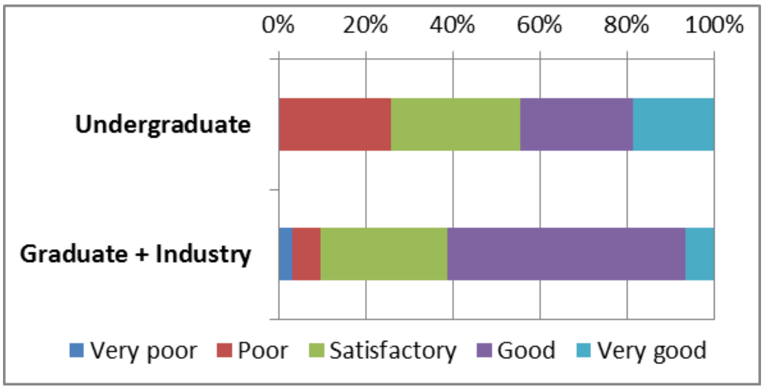


Figure 6.14: Assessments by different groups of participants on how well the documents were in helping them to identify the information that they might need to answer questions in Part 1 (P2Q5a).

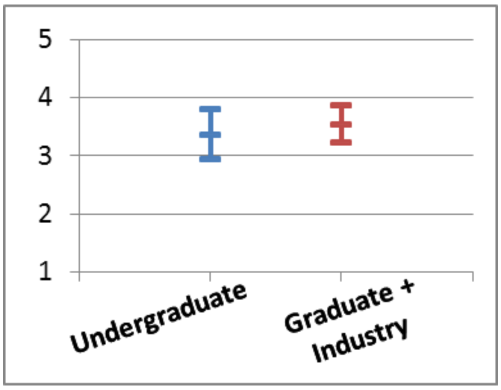


Figure 6.15: 95% confidence intervals for the responses to P2Q5a by different groups of participants.

To test for any significant difference in the means between Undergraduate participants and Graduate + Industry participants, we also perform an independent-samples t-test. The null hypothesis for the test is that there is no difference between the two means. The t-test gives a two-tailed p-value of 0.486 which is more than the 0.05 level of significance (t = 0.49, df = 56, mean difference = −0.18, 95% confidence interval of difference = −0.69, 0.33). Therefore, we do not reject the null hypothesis. This suggests that there is no significant difference in the means. The statistical analysis suggests that our participants' assessments of the accessibility of the ReqSpec document and the UCM document was not affected by their experience. Furthermore, Cohen's effect value (d = 0.19) suggests low practical significance.

The statistical analysis shows no statistically significant difference in the accessibility of information between Undergraduate participants and Graduate + Industry participants. The difference is also not practically significant. The experience of our participants did not affect our participants' ability in identifying information to answer the questionnaire.

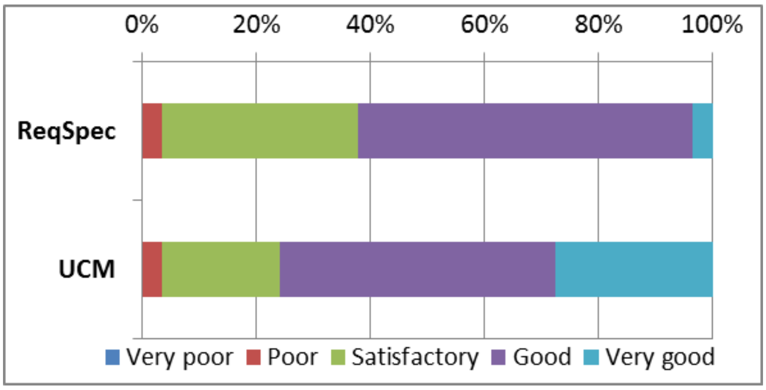


Figure 6.16: Participants' assessments of the helpfulness of the ReqSpec document and the UCM document to understand the functionality of the software system (P2Q5c).

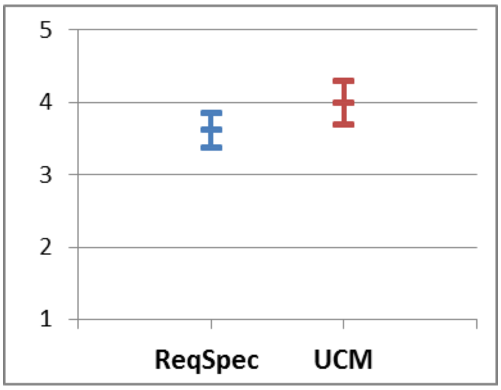


Figure 6.17: 95% confidence intervals for the responses to P2Q5c by participants using the ReqSpec document and the UCM document.

Understandability of Information

In Part 2 of the questionnaire, we ask participants how helpful they think that the documents are to understand information and how well they think that they have understood the information in the documents. Figure 6.16 shows participants' assessments on the ReqSpec document and the UCM document in helping them to understand the functionality of the software system (P2Q5c).

More than 60% of our participants reported that both documents were good or very good in helping them to understand the functionality of the software system. Two out of the 58 participants reported that the documents were poor.

The mean values for treatment ReqSpec and treatment UCM are 3.62 and 4.00 with standard deviations 0.62 and 0.80 respectively. A comparison between the two 95% confidence intervals suggests that there might be no significant difference in understanding information using different documents as illustrated in Figure 6.17.

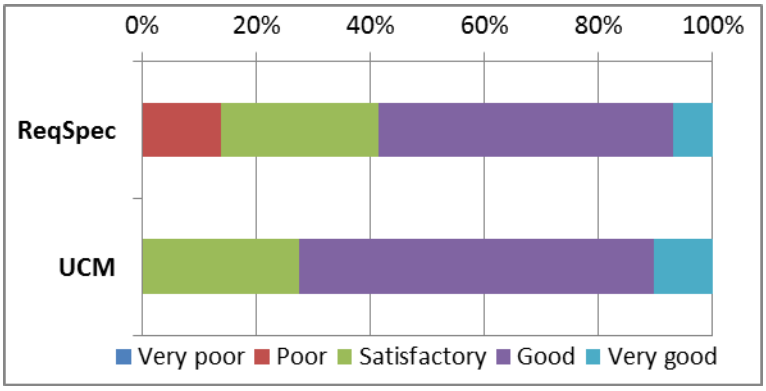


Figure 6.18: Participants' self-assessments on how well they have understood the information provided in the ReqSpec document and the UCM document (P2Q6a).

We perform an independent-samples t-test for any significant difference in the means. The null hypothesis is that there is no significant difference in the means for treatment ReqSpec and treatment UCM. The t-test gives some evidence against the existence of no difference between the means ($p = 0.049$). The t-value is -2.01 with 56 degrees of freedom. The mean difference is -0.38, and the 95% confidence interval of the difference is -0.76 and -0.002. Moreover, Cohen's effect value ($d = 0.53$) suggests a moderate practical significance. Since the mean for the UCM document is greater than the mean for the ReqSpec document, the UCM document is more helpful than the ReqSpec document in participants' understanding of the functionality of the software system.

In P2Q6a, we ask a similar question about how well participants think that they have understood the information provided in the documents. As shown in Figure 6.18, more than half of the 58 participants reported that they have a good or very good understanding of the documents. No participants reported that they understood the information poorly except for four participants who used the ReqSpec document.

We plot a 95% confidence interval graph to compare the means as shown in Figure 6.19. It seems that there might not be any significant difference in participants' self-assessments on how well they understood information in the documents as the two confidence intervals do not overlap. The means are 3.52 and 3.83 with standard deviations 0.83 and 0.60 for treatment ReqSpec and treatment UCM respectively.

We perform an independent-samples t-test to test the null hypothesis: there is no difference in the means for how well participants understood information using the ReqSpec document and the UCM document. The t-test shows a two-tailed p-value of 0.109 which suggests that there is no significant difference at the 0.05 level of significance ($t = -1.63$, $df = 51.09$, mean difference $= -0.31$, 95% confidence interval of difference $= -0.69, 0.07$). Cohen's effect value is 0.43 which suggests a low to moderate practical significance.

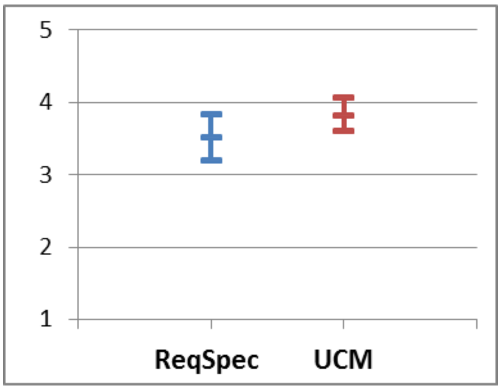


Figure 6.19: 95% confidence intervals for the responses to P2Q6a by participants using the ReqSpec document and the UCM document.

The statistical analysis shows that there is some evidence against the null hypothesis. It seems that the UCM document is better than the ReqSpec document for the understandability of functional requirements. However, the statistical analysis for P2Q6a shows no significant difference in the understandability of information using the ReqSpec document and the UCM document by our participants. The mean values from P2Q5c and P2Q6a indicate that both documents were more than satisfactory in helping participants to understand information. The analysis gives us some evidence to support our analysis of understandability of the requirements documents as discussed in Section 6.1.5.

Understandability of Information by Experience

Figures 6.20 and 6.21 display the results of different groups of participants' assessments of the understandability of information. In P2Q5c, the mean values for Undergraduate participants and Graduate + Industry participants are 3.93 and 3.71 with standard deviations of 0.68 and 0.78 respectively. In P2Q6a, the mean values for Undergraduate participants and Graduate + Industry participants are 3.74 and 3.61 with standard deviations of 0.71 and 0.76. It seems that the requirements documents were more than satisfactory for both groups of participants in helping to understand information. Participants' experience did not affect the assessments of the understandability of information.

To compare the differences in the means, we plot two confidence interval graphs for P2Q5c and P2Q6a as shown in Figures 6.22 and 6.23. The graphs show that there might not be any significant differences between the means as the confidence intervals overlap. To establish any significant difference between the means, we perform an independent-samples t-test to test the null hypothesis that there is no difference between the means for P2Q5c. Similarly, we perform an independent-samples t-test for P2Q6a. The t-test shows p-value of 0.268 for P2Q5c (t = 1.12, df = 56, mean difference = 0.22, 95%

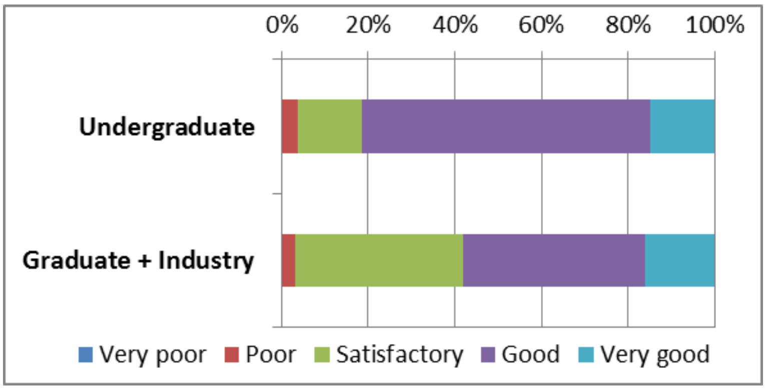


Figure 6.20: Assessments by different groups of participants on how well the documents were in helping them to understand the functionality of the software system (P2Q5c).

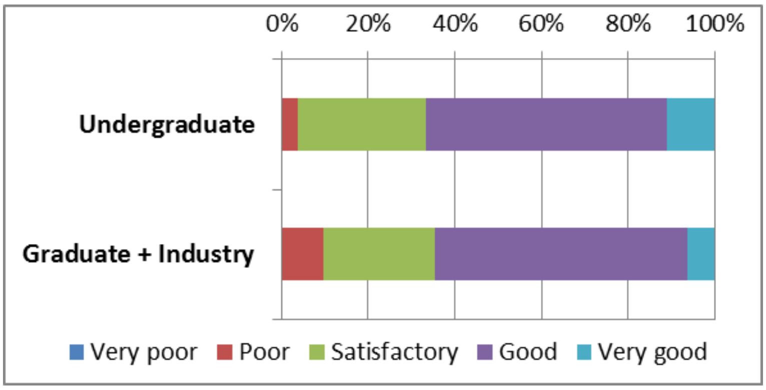


Figure 6.21: Assessments by different groups of participants on how well they have understood the information provided in the requirements documents (P2Q6a).

confidence interval of difference = −0.17, 0.60). For P2Q6a, the t-test shows p-value of 0.513 ($t = 0.66$, $df = 56$, mean difference = 0.13, 95% confidence interval of difference = −0.26, 0.52). These tests also indicate that there are no statistically significant differences at 0.05 level of significance. Further, Cohen's effect value for P2Q5c is 0.03 and for P2Q6a is 0.18, which both suggest low practical significance.

The statistical analysis suggests that our Undergraduate participants and Graduate + Industry participants have similar ratings for how well the documents were in helping them to understand information. Both groups of participants also have similar ratings for how well they have understood the information provided in the documents.

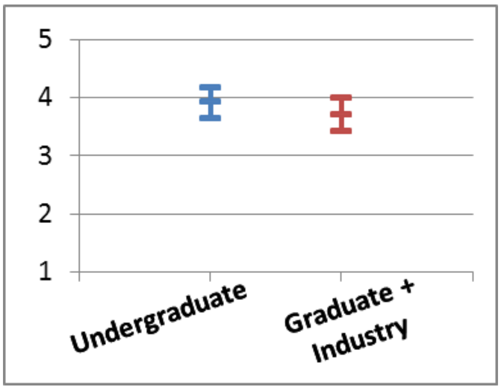


Figure 6.22: 95% confidence intervals for the responses to P2Q5c by different groups of participants.

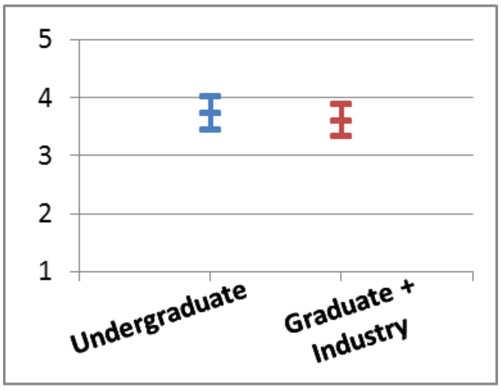


Figure 6.23: 95% confidence intervals for the responses to P2Q6a by different groups of participants.

Relevance of Information

As discussed in Section 6.1.5, we believe that the UCM document provides more relevant information to our participants for answering questions in Part 1 of the questionnaire than the ReqSpec document. In Part 2 of the questionnaire, we ask two questions about the relevance of information.

We first ask participants in P2Q4 whether they have to go through different parts of the requirements documents in order to answer P1Q6. P2Q4 enables us to evaluate the sufficiency of the information at a particular location to answer the questions. If the information is insufficient, participants are likely to try and look for another location in the document.

Table 6.10 shows the number of participants who either went through different parts of the document or not. It appears that there were more participants who went through different parts of the ReqSpec document than participants who went through the UCM document to answer P1Q6. The observed proportion of yes to no for participants using

	ReqSpec	UCM	Total
Yes	20	11	31
No	9	18	27
Total	29	29	58

Table 6.10: Number of participants who either went through different parts of the requirements document to answer P1Q6 (P2Q4) or not.

the ReqSpec document is 0.69:0.31, whereas the proportion of yes to no for participants using the UCM document is 0.38:0.62. We compare the two proportions by using Fisher's exact test. The null hypothesis for the test is that there is no difference between the two proportions. We get a two-tailed p-value of 0.03 which is significant at the 0.05 level. Therefore, we reject the null hypothesis. This supports the existence of a difference between participants using different documents to answer P1Q6. In addition, the Phi coefficient of association ($\phi = -0.31$) suggests a weak negative association.

We also ask participants in P2Q5b to rate how helpful they think that the documents are to read only the relevant information to answer questions in Part 1. Figure 6.24 shows the distribution of participants' assessments on the requirements documents in P2Q5b. Participants using the ReqSpec document seem to have varied opinions about the document. On the other hand, approximately 80% of participants using the UCM document reported the UCM document was good or very good in reading relevant information.

The means for the responses by participants using the ReqSpec document and participants using the UCM document are 2.97 and 3.76 with standard deviations of 1.12 and 0.95 respectively. We plot a 95% confidence interval graph as illustrated in Figure 6.25 to compare the means. There is a difference in how well the documents helped participants to read relevant information as the two confidence intervals do not overlap with each other. We perform an independent-samples t-test to test the null hypothesis that there is no difference between the means. We find that the two-tailed p-value is 0.005 from the t-test which is less than the 0.05 level of significance ($t = -2.91$, $df = 56$, mean difference $= -0.79$, 95% confidence interval $= -1.34, -0.25$). Hence, we reject the null hypothesis. This indicates that there is a significant difference in the relevance of information in the ReqSpec document and the UCM document. Furthermore, Cohen's effect value ($d = 0.76$) suggests a moderate to high practical significance.

The analysis shows that the UCM document provides more relevant information than the ReqSpec document. Fewer participants who used the UCM document went through different parts of the document than participants who used the ReqSpec document. Participants who used the UCM document tended to be more satisfied with the relevant information than participants who used the ReqSpec document.

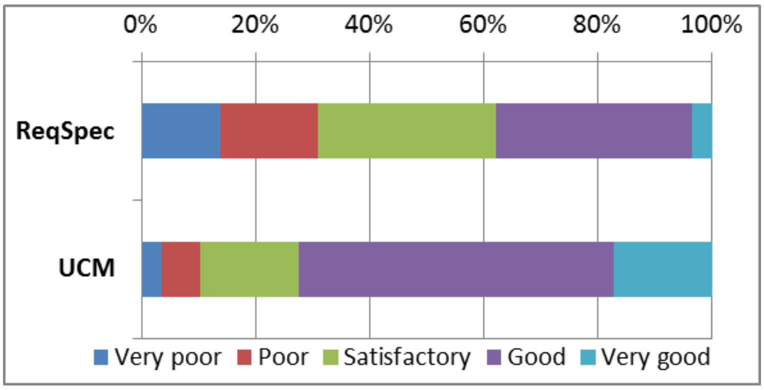


Figure 6.24: Participants' assessments on how well the ReqSpec document and the UCM document were in helping participants to read only the relevant information that they needed to answer each question in Part 1 (P2Q5b).

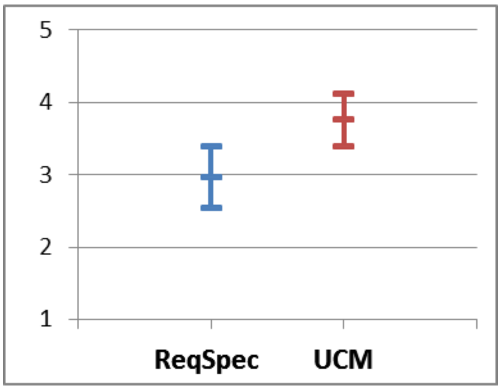


Figure 6.25: 95% confidence intervals for the responses to P2Q5b by participants using the ReqSpec document and the UCM document.

Relevance of Information by Experience

Table 6.11 shows the number of participants with different experience who went through different parts of the requirements documents. The number of Undergraduate participants who answered yes to P2Q4 is almost the same as the number of Graduate + Industry participants who answered yes to P2Q4. The observed proportion of yes to no for Undergraduate participants is 0.56:0.44. The observed proportion of yes to no for Graduate + Industry participants is 0.52:0.48. We perform the Fisher's exact test for the null hypothesis that there is no difference between the two proportions. The test gives us a two-tailed p-value of 0.80 which suggests no significant difference between the two proportions. Therefore, we do not reject the null hypothesis. In addition, the Phi coefficient of association ($\phi = -0.04$) suggests little or no association.

Figure 6.26 shows how helpful participants found the requirements documents in reading only the relevant information to answer Part 1 of the questionnaire. The means for

	Undergraduate	Graduate + Industry	Total
Yes	15	16	31
No	12	15	27
Total	27	31	58

Table 6.11: Number of participants in different groups went through different parts of requirements documents in order to answer P1Q6 (P2Q4).

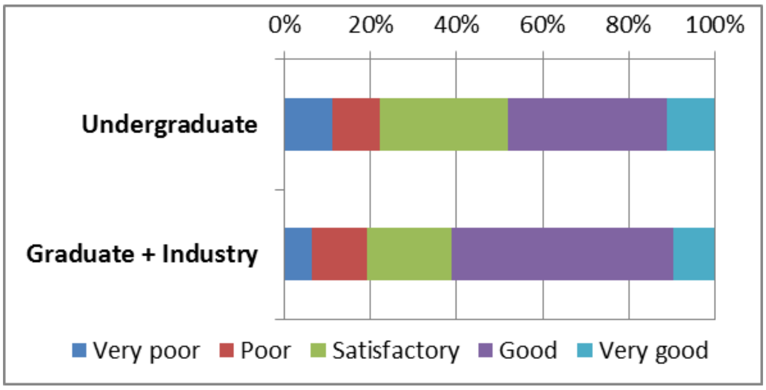


Figure 6.26: Assessments by different groups of participants on how helpful the requirements documents were in reading only the relevant information to answer each question in Part 1 (P2Q5b).

the responses by Undergraduate participants and Graduate + Industry participants are 3.26 and 3.45 with standard deviations of 1.16 and 1.06 respectively. It seems that both groups of participants rated the documents more than satisfactory for the relevance of information.

To compare if there is any difference between the means, we plot a 95% confidence interval graph as illustrated in Figure 6.27. The confidence intervals overlap which suggests that there might be no significant difference. We perform an independent-samples t-test for the null hypothesis that there is no difference between the two means. The test shows that there is no significant difference between the two means as the two-tailed p-value is 0.51 at 0.05 level of significance ($t = -0.66$, $df = 56$, mean difference = -0.19, 95% confidence interval = -0.78, 0.39). Further, Cohen's effect value ($d = 0.17$) suggests low practical significance.

The statistical analysis suggests that the relevance of information is not affected by the experience of our participants. The analysis shows that no significant difference in the number of Undergraduate participants and the number of Graduate + Industry participants who went through different documents. We also find that the two groups of participants have similar ratings for how helpful the documents were in reading relevant information.

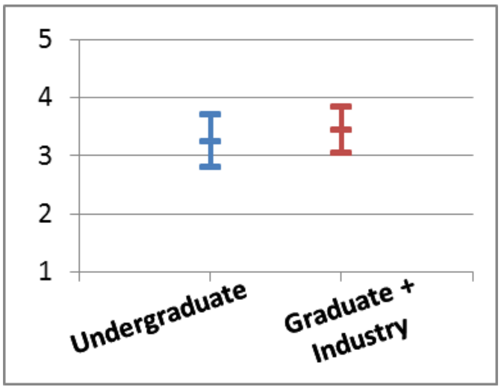


Figure 6.27: 95% confidence intervals for the responses to P2Q5b by different groups of participants.

Hypothesis Testing: Confidence of Participants (EH3)

To test EH3, we ask participants in P2Q6b for their confidence in answering the questions in Part 1 correctly. Figure 6.28 shows the distribution of participants' self-assessments on question P2Q6b. Of the 29 participants in treatment ReqSpec, six participants answered good for Part 1. On the other hand, of the 29 participants in treatment UCM, 18 reported good or very good for their answers to Part 1. The mean responses for the ReqSpec document and the UCM document are 2.76 and 3.62 with standard deviations 0.87 and 0.73 respectively. It seems that participants using the UCM document felt more confident than participants using the ReqSpec document about the correctness of their answers.

As illustrated in Figure 6.29, the two confidence intervals do not overlap suggesting a difference in participants' using different documents. To test for any statistically significant difference, we perform an independent-samples t-test with the null hypothesis that there is no difference between the mean responses. The t-test shows that there is a significant difference as the two-tailed p-value is 0.00. The t-value we obtain is -4.09 with 56 degrees of freedom. Therefore, we reject the null hypothesis.

By examining the two means and the mean difference, we see that participants using the ReqSpec document were less confident than participants using the UCM document. The mean difference is -0.86. The difference in the 95% confidence level is between -1.29 and -0.44. We also calculate the effect size using Cohen's d, which suggests a high practical significance ($d = 1.07$).

The analysis gives us high confidence to support EH3: there is significant difference in the confidence of participants using different requirements documents. Since the UCM document mean is greater than the ReqSpec document mean, participants were more confident with their answers using the UCM document than participants using the ReqSpec document.

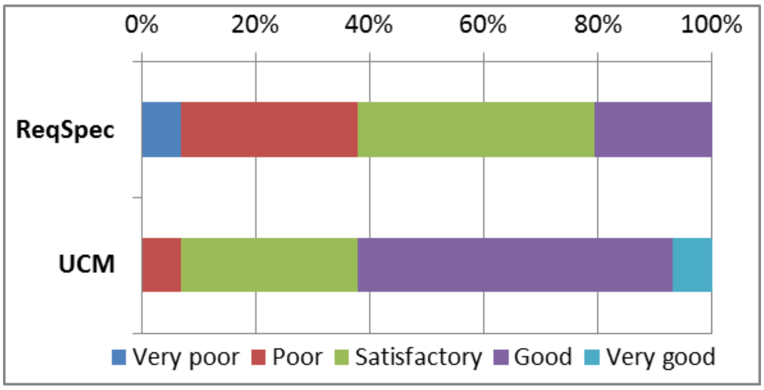


Figure 6.28: Participants' self-assessments on their confidence in answering Part 1 questions correctly using the ReqSpec document and the UCM document (P2Q6b).

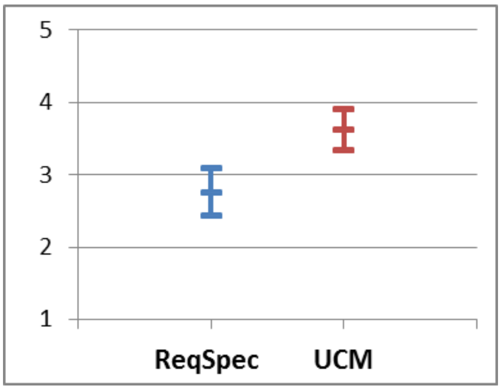


Figure 6.29: 95% confidence intervals for the mean responses to P2Q6b by participants using the ReqSpec document and the UCM document.

Confidence of Participants by Experience

Figure 6.30 shows the distribution of participants' self-assessments on question P2Q6b. More than 50% of participants in each group reported satisfactory, good or very good to P2Q6b. The mean value for Undergraduate participants is 3.15 with a standard deviation of 1.03. The mean value for Graduate + Industry participants is 3.23 with a standard deviation of 0.81. Both groups of participants seem to be confident in their answers to Part 1 of the questionnaire. It appears that there is no difference between the two groups of participants.

The 95% confidence interval graph in Figure 6.31 shows that there might be no difference in the mean values between different groups of participants. We compute an independent-samples t-test for comparing the two groups of participants. The null hypothesis is that there is no difference in the mean values between Undergraduate participants and Graduate + Industry participants. The t-test gives a two-tailed p-value of 0.748 which is not significant at the 0.05 level of significance. The t-value is −0.32, with

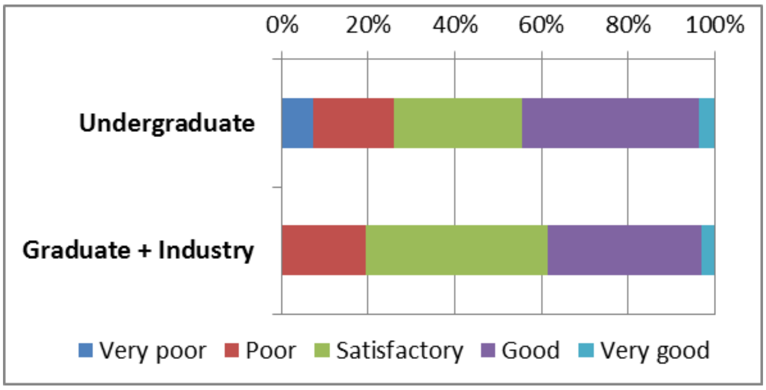


Figure 6.30: Assessments by different groups of participants on confident they felt in answering Part 1 questions correctly (P2Q6b).

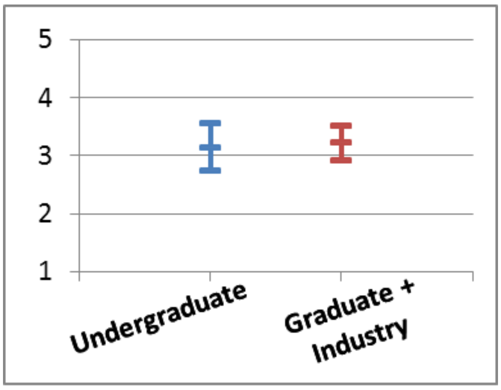


Figure 6.31: 95% confidence intervals for the responses to P2Q6b by different groups of participants.

56 degrees of freedom. The mean difference is 0.08, and the 95% confidence interval of the difference is −0.56 and 0.41. Therefore, we cannot reject the null hypothesis. Further, Cohen's effect size value ($d = 0.09$) suggests low practical significance.

The statistical analysis shows no difference in the confidence level of Undergraduate participants and Graduate + Industry participants. This suggests that the experience of our participants did not affect how confident participants felt in giving correct answers.

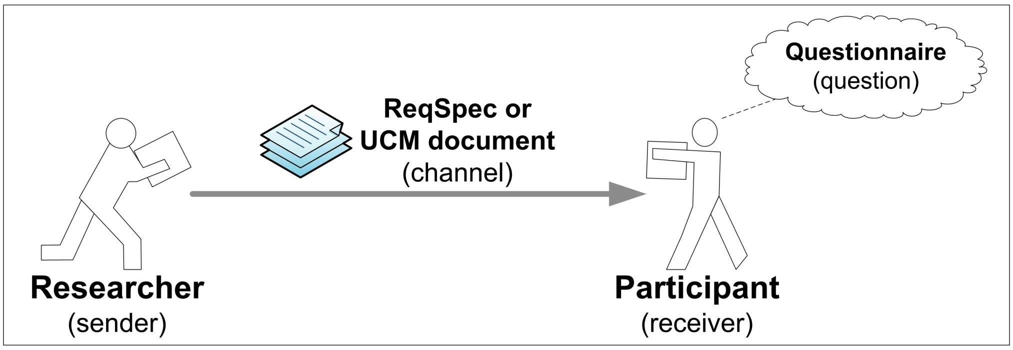


Figure 6.32: A simple communication model in the context of the experiment.

6.3.4 Themes

Our experiment involves the researcher as the sender of information and participants as the receiver of information. As shown in Figure 6.32, the channel is the ReqSpec document or the UCM document. The questionnaire for the experiment is used as the set of questions for a receiver to answer. In our experiment, we are interested in the transparency of channel in presenting functional requirements of a software system to the receiver. We are interested in exploring the following questions from the participants' comments:

- What affects the assessment of transparency of the communication channel?
- What affects the accessibility, understandability, and relevance of the communication channel?
- Are there any other interesting themes related to communication that emerged from the experiment?

In this section, we discuss the themes that emerged from the participants' comments. The themes are organised according to our attributes of transparency. Some themes identified from the experiment have sub-themes, which can be positive or negative comments about software artefacts in general and the documents used in the experiment. There are also comments which we coded as neutral themes. Figure 6.33 illustrates the basic elements of a theme. In addition, we include the total number of participants who commented on each theme.

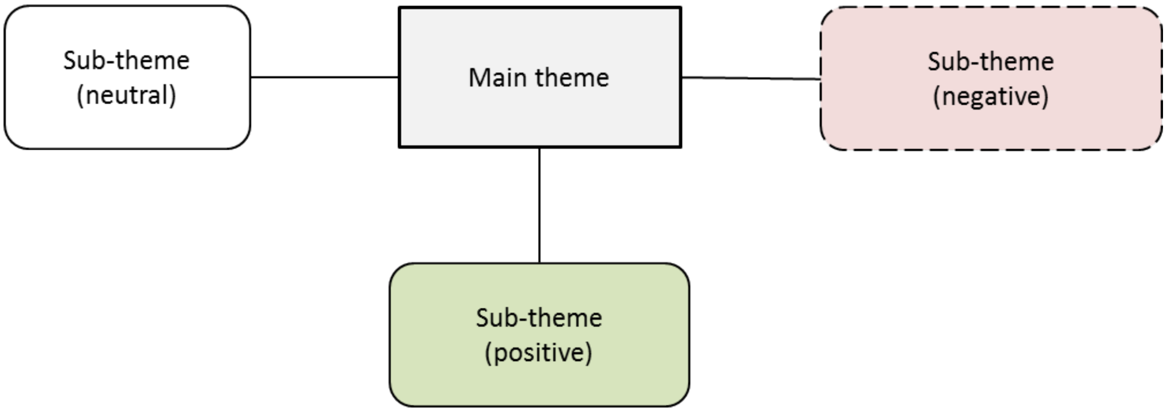


Figure 6.33: Basic elements of a theme.

Factors Affecting Assessment of Transparency

In this section we discuss themes that affect the assessment of a communication channel's transparency from the participants' comments. We find four main themes as summarised in Figure 6.34. These themes are concerned with the assumptions that we made about our working definition of transparency.

The first theme is "transparency of information is context dependent". We identify two sub-themes that are related to this theme. The first sub-theme is about the relevance of information. We discover the first sub-theme from the comments made by participants on the effectiveness of different software documents or models in the demographic section. According to one of the 58 participants, software documents' effectiveness in helping participants to understand functional requirements depends on "the type of project [that they] are working on". The participant further explained that "[f]or a small project a simple informal statement of work should be enough". The comment suggests that information required by stakeholders to understand requirements depends on the type of project. This, in turn, suggests that the relevance of information depends on the context in which the receiver is situated.

The second sub-theme is related to the understandability of information. This theme comes from participants' comments on concerns about software artefacts in general. A participant commented on "finding the 'right way' to describe software functionalities & descriptions to the client which suits their knowledge / usual approach of describing, so they may better understand about the system". Another participant mentioned that "less technical documents would be useful for communication with stakeholders outside of SE [(Software Engineering)]". The comments show that information's presentation is important to help the receiver to understand information. Moreover, the comments show that how well the receiver understands information depends on the receiver's role.

Therefore, understandability of information depends on the context in which the receiver is situated.

Another main theme affecting a receiver's assessment of the quality of channel's transparency is related to the receiver's expectation of the communication channel. In our experiment, the communication channel is either the ReqSpec document or the UCM document. It is possible that our participants (receivers) have unreasonable expectations for the quality of information presented in the requirements documents (communication channel). Although there are no "unreasonable" comments in the experiment, one participant seemed to have a negative feeling regarding the ReqSpec document in saying it "was a little annoying not being functionally driven". This comment indicates the participant's expectation that the structure of ReqSpec document be functionally driven. However, it is unclear how the negative feeling affected the assessment of transparency.

The third theme is related to the receiver's prior knowledge and experience. As discussed in Chapter 3, the receiver's prior knowledge or experience affects how much time the receiver expects to spend on obtaining and assessing information to answer his or her questions. In our experiment, we identify three sub-themes related to this main theme. The first sub-theme is a positive comment on the effectiveness of software documents in the demographic section. The participant said it is "easy to find information [within] if you are familiar with the standard", referring to requirements documents that follow a specific format or standard. This comment implies that a receiver can easily access information in a requirements document if the receiver knows the document's format or standard. This in turn suggests that the receiver's familiarity with the standard can improve the accessibility of information.

The second and third sub-themes are negative themes that we find from comments on the ReqSpec document and the UCM document. Our participants commented on the problems that they encountered when answering Part 1 of the questionnaire. Some of our participants believed that their lack of knowledge or experience could affect how they understood information provided in the requirements documents. For example, one participant commented that the ReqSpec document "... was a bit confusing to begin with, but I think this is because of my lack of exposure to such document, and not the document ..." Another participant commented that he or she could not finish reading the UCM document due to "poor English skill". The comments suggest that a receiver's limited knowledge or experience affects the time it takes for the receiver to understand information. In the experiment, one participant could not answer questions in Part 1 using the ReqSpec document - the participant was not too sure "how to acquire the relevant information for a question". The question that the participant could not answer was P1Q7, one that many participants had trouble answering as discussed previously. The comment made by participants shows that the ability of a receiver in answering questions can be affected by the receiver's uncertainty in looking for information.

The last theme affecting the assessment of transparency is related to the questions that a receiver has in a simple communication model. In Chapter 3, we discussed our assumption about the questions that a receiver has. The type of question depends on the receiver's preconceived ideas of questions and finding answers. Their questions also affect how the receiver finds desired information. This in turn affects the accessibility of information. In our experiment, the questionnaire is used as the set of questions that a receiver has to answer. We refer to the questions in the questionnaire as the researcher's questions. We find several comments made by participants on how the researcher's questions affected participants' ability to find information. Firstly, participants' interpretations of the questions affect how participants access information. It seems that one participant could not find information as he or she could not understand the questions. Another participant could not find information relevant to the questions. On the other hand, one of the participants commented that he or she could find the information easily because "the titles in the questionnaire matched the titles used in the document". This suggests that accessibility of information is affected by a receiver's interpretation of questions. It also depends on the context, which in our experiment means the questionnaire.

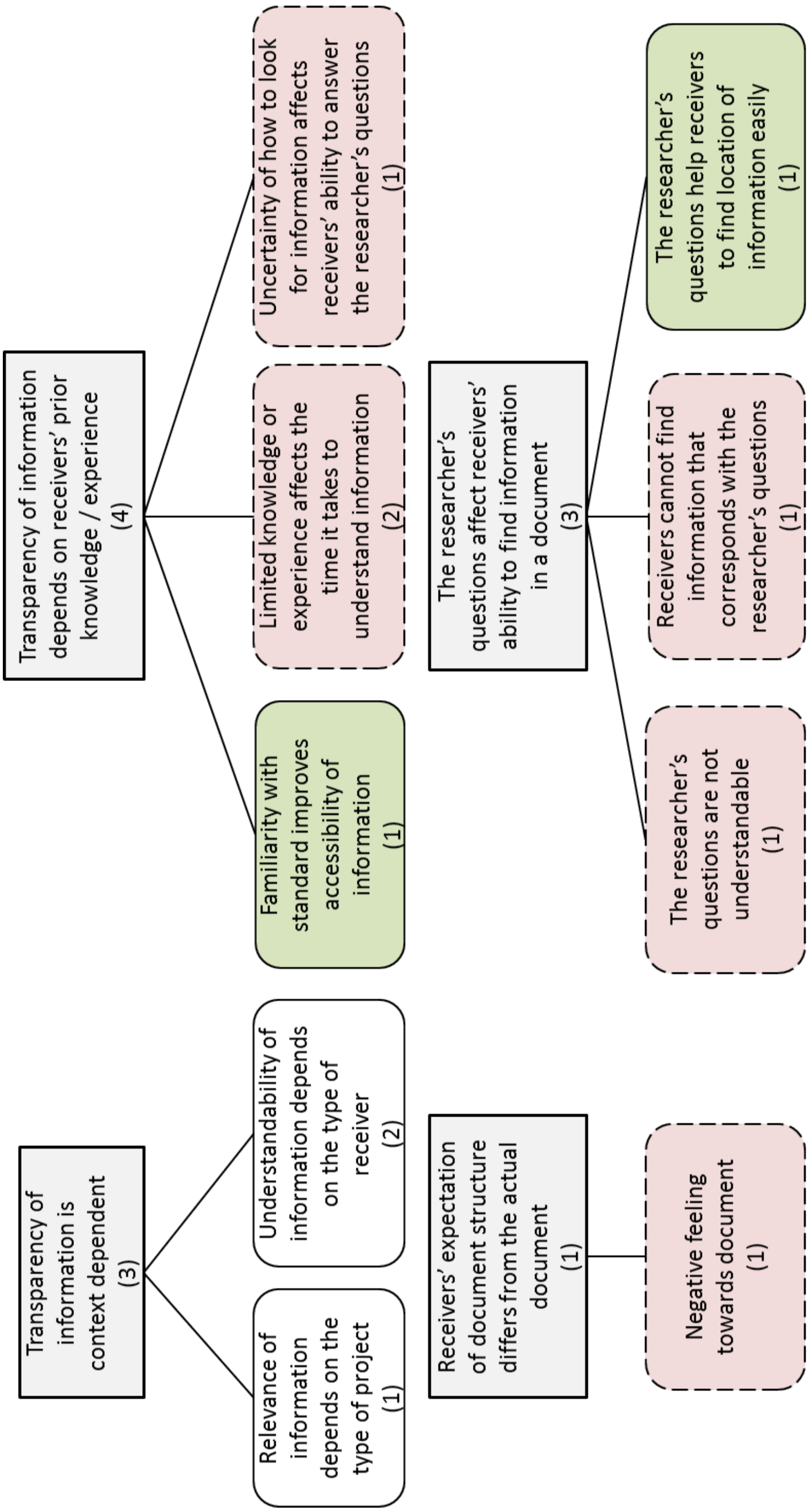


Figure 6.34: Themes that affects the assessment of transparency.

Accessibility

We find seven themes that affect the accessibility of information as summarised in Figure 6.35. The first theme concerns how the sender of information affects accessibility. Participants commented on concerns regarding the sender as well as software artefacts. These concerns indicate that the accessibility of information might be hindered by the sender's willingness to share information. The accessibility of information might also be affected by a lack of software artefacts.

In the experiment, we find that the organisation of the ReqSpec document and the UCM document have positive and negative effects on the accessibility of information. For example, participants found the use case diagram, the document structure, and the table of contents in the UCM document helpful in locating information. On the other hand, participants using the ReqSpec document commented that headings and sections of the document needed to be improved. An index and an appendix could be included in the ReqSpec document to improve participants' locating information.

Another theme that arises from the experiment is the format of the document. Most of our participants were given physical copies of the ReqSpec document and the UCM document in the experiment. Participants were required to find information in the document manually, which in turn could take more effort than searching for information electronically. A few of our participants made that observation. One participant also commented that his or her "ability to manually search text has diminished" because he or she became used to finding information on a computer. It seems that information in electronic format could help to improve accessibility of information.

We also find different factors that hinder participants in locating information within the ReqSpec document or the UCM document. For example, participants using the ReqSpec document found similar information was distributed throughout the document, and as a result they were confused when trying to locate specific information. Some participants using the ReqSpec document also commented that the table of contents was not helpful for finding information or that the navigation of the document was not easy. Similarly, one participant using the UCM document mentioned that he or she needed to "... refer back and forth..."

Among the comments made by participants using the ReqSpec document, there is a common theme regarding time. Out of all 58 participants, five participants who used the ReqSpec document noted that they could not locate the information after spending 10 minutes or a long time on each question of Part 1 of the questionnaire. However, we did not find any participants who used the UCM document commenting that they spent more than 10 minutes on each question. Similarly, at least 10 of the 29 participants using the

ReqSpec document mentioned that they needed to look through the document to answer questions whereas no participants using the UCM document made that comment.

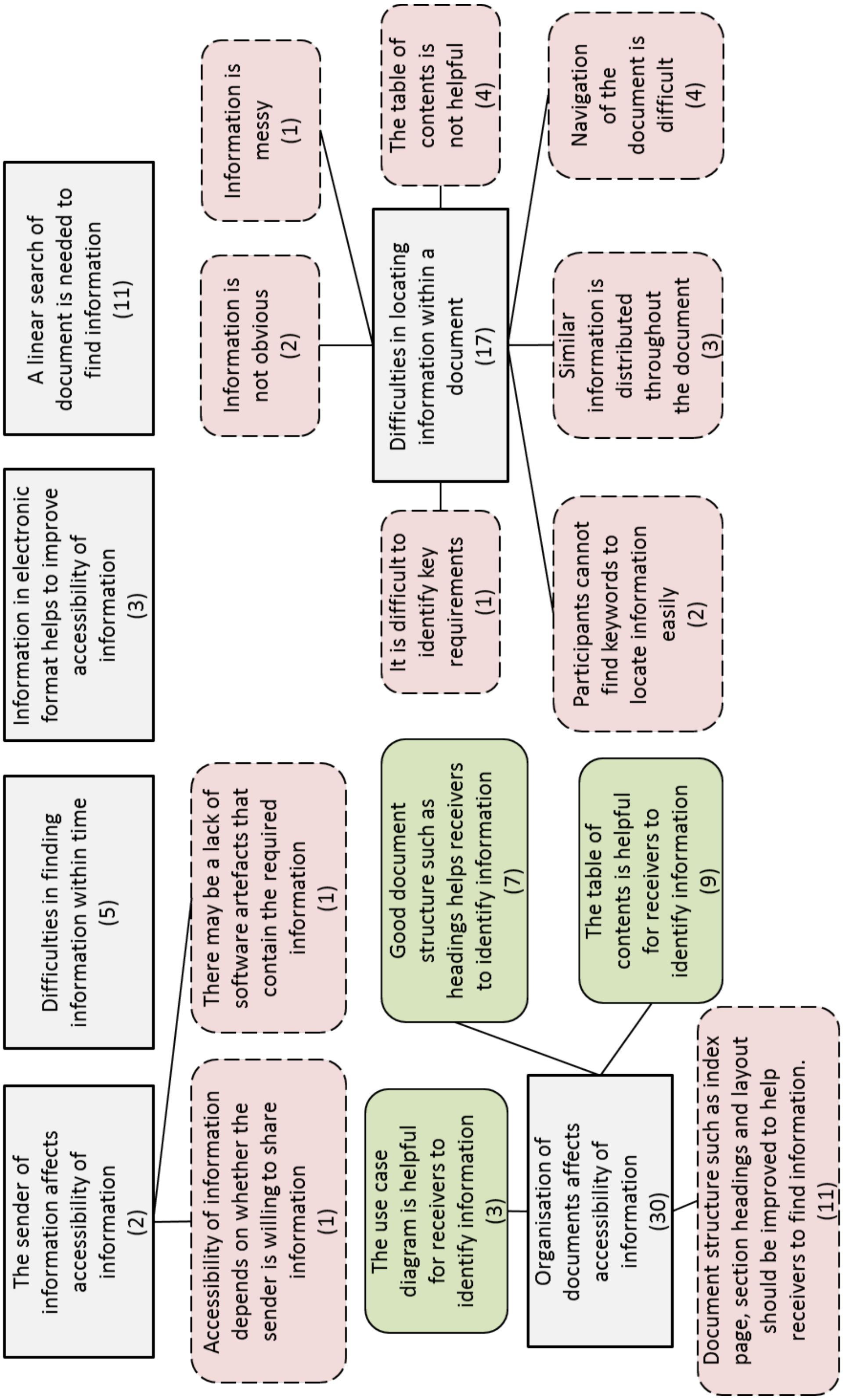


Figure 6.35: Themes for the "Accessibility" attribute of transparency.

Understandability

In the experiment we identify three main themes that are related to the understandability of information. Figure 6.36 summarises the three understandability themes.

The first theme is related to the sender of information, in which the sender affects the presentation of information. This in turn affects how well the receiver understands this information.

The second theme is related to how the ReqSpec document and the UCM document affect the understandability of information. According to our participants who used the UCM document, the use case diagram was useful in helping them to understand the functionality of the system. However, a few of the participants who used the UCM document suggested that the use case diagram was insufficient. More diagrams like workflow diagrams could improve understanding of the system's functionality. Participants using the ReqSpec document also suggested including use case diagrams as well as diagrams like sequence diagrams in the document to help readers understand the system. Similarly, participants using the ReqSpec document and participants using the UCM document suggested that using pictures or illustrations helps in understanding.

The third theme is related to different factors that hinder participants' understanding of the information. A few of our participants commented that they needed more time to understand the information presented, particularly in the ReqSpec document. Similarly, the terminology and abbreviations used in the ReqSpec document and the UCM document were not easy for two of our participants. Another factor that hindered the participants' understanding of information is the confusing nature of the information in the ReqSpec document. Of the 58 participants, four participants commented that the information was confusing.

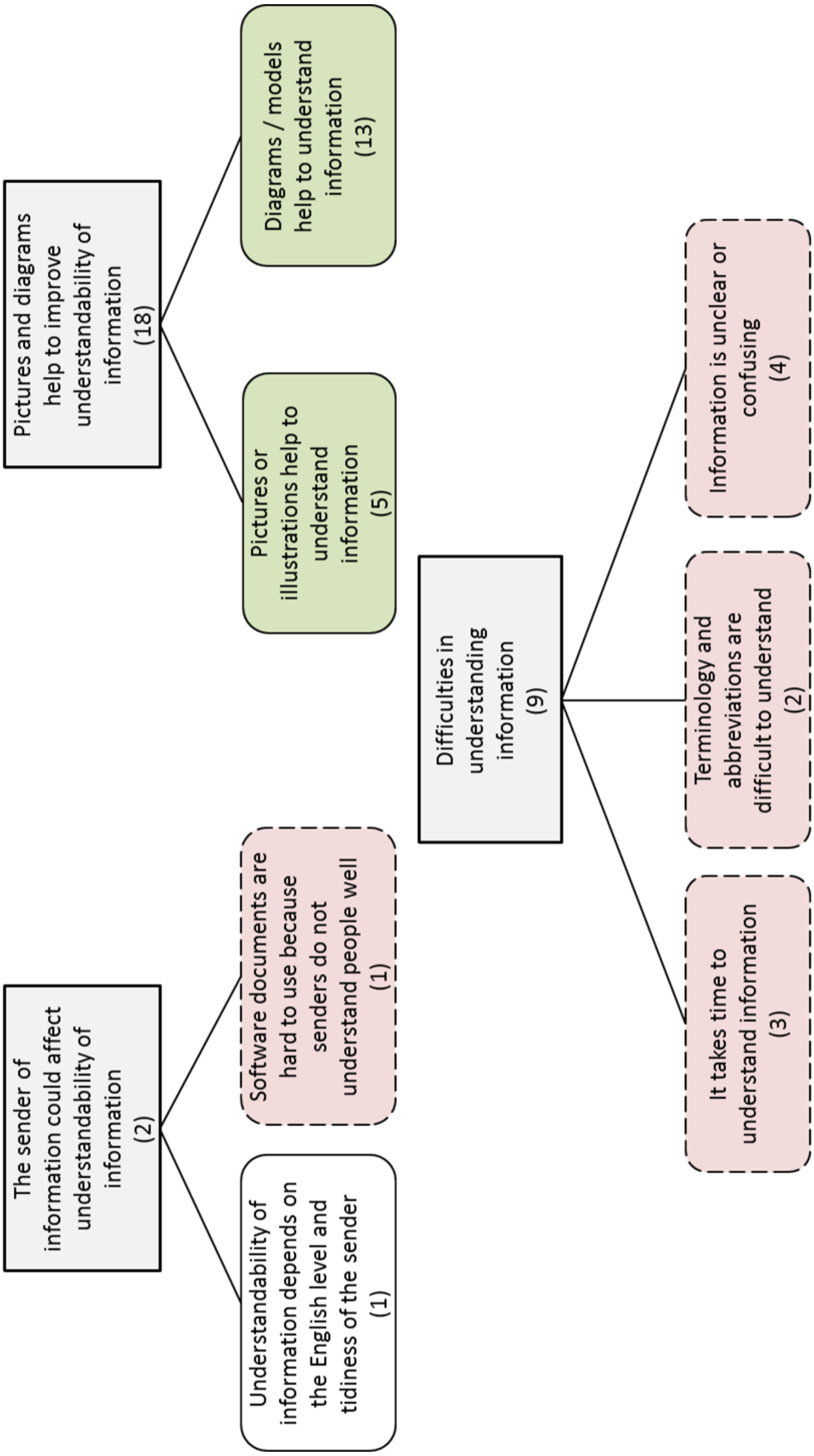


Figure 6.36: Themes for the "Understandability" attribute of transparency.

Relevance

We find three main themes from the experiment that affect how relevant receivers thought the information was to answer the questions. Figure 6.36 shows the summary of the three themes conveying the relevance of the information. In the experiment, we find 26 participants commented that they could not answer questions sufficiently using the requirements documents. Participants commented on problems such as missing detailed information in the documents. Participants also commented on the information in the documents being unclear which also affected their ability in understanding information. In addition, there were comments regarding question P1Q7. Participants commented that they could not find clear information or direct quotes to answer the question.

The second theme that we find is related to receivers having too much information which might affect the time that receivers spend on answering their questions. Several participants using the ReqSpec document reported that there was too much text to read in the document. There were also two participants using the UCM document who reported that the use cases were long. Furthermore, there were concerns about over-documentation and long documents which could cause participants to spend too much time on documenting or reading irrelevant information.

Another theme is related to the problem of finding relevant information by our participants. This theme comes mainly from the responses made by participants to P1Q7. Twenty-three participants commented that they could not find the information at the expected location to answer P1Q7. For example, one of the participants who used the ReqSpec document reported that he or she "looked in section 3 page 27 because contents suggested data requirements but did not find relevant information." Similarly, a participant who used the UCM document answered P1Q7 with the comment: "... not seen relevant information on page 17. Neither for the Use Case Diagram on page 4." Based on such comments, we find that the information presented in the documents could be irrelevant for answering questions.

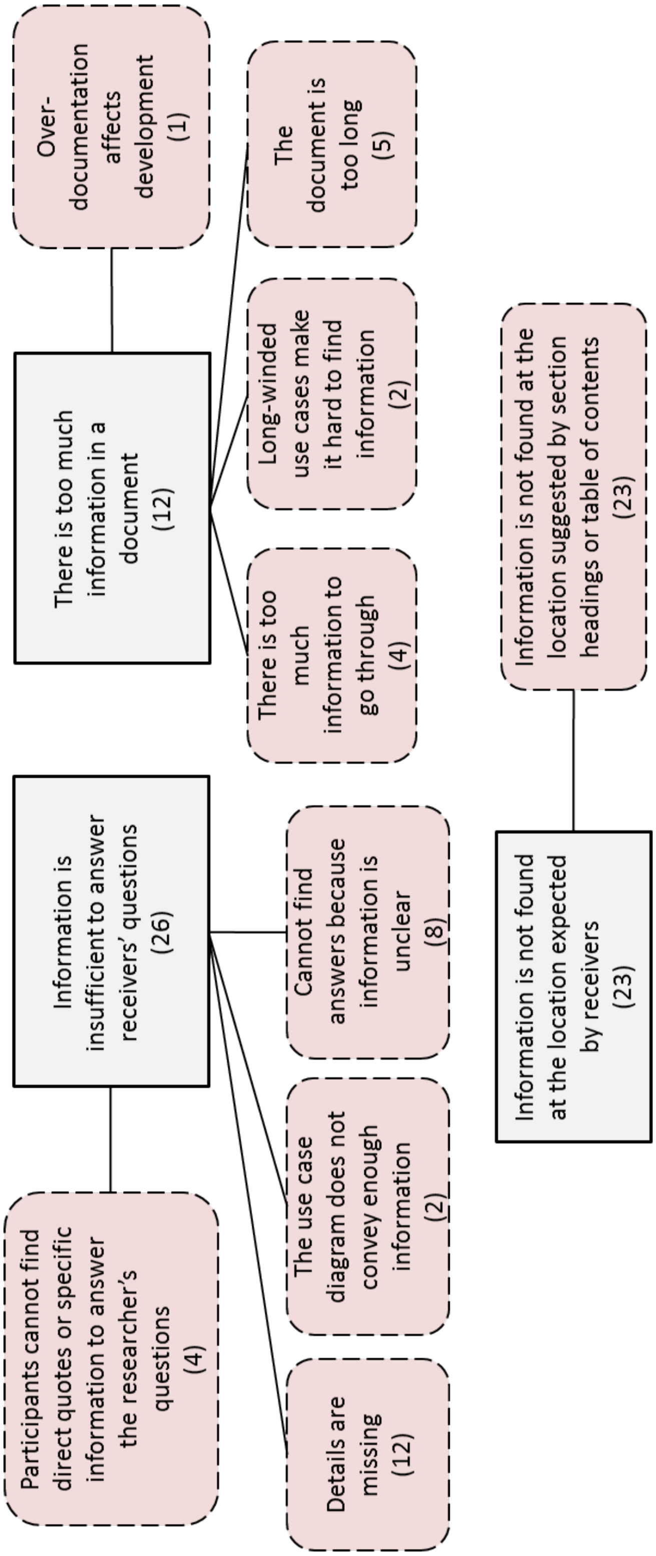


Figure 6.37: Themes for the "Relevance" attribute of transparency.

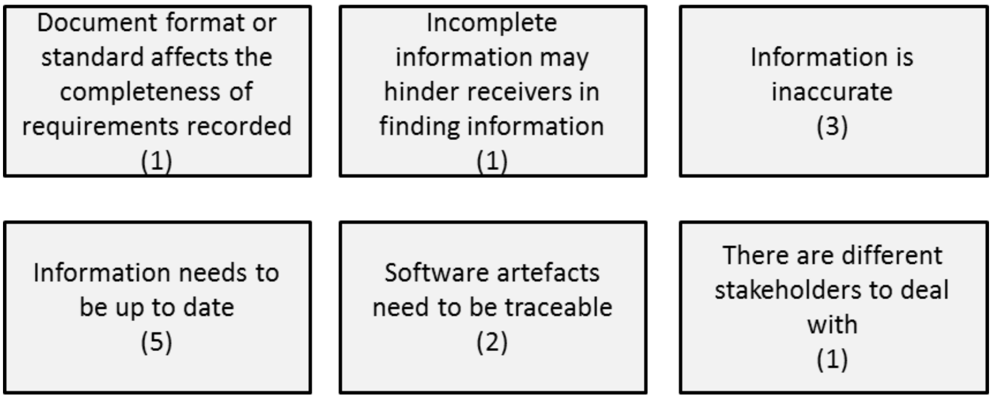


Figure 6.38: Interesting additional themes that arise from the experiment.

Other Concerns

We find six interesting themes that arise mostly from the comments made to P2Q9 (concerns about software artefacts or communication in general) in the experiment, as summarised in Figure 6.38. We find these themes affect communication between the sender and the receiver of information.

The first two themes about document format and incomplete information are related to the completeness of information. Firstly, one of the participants commented that "following standards ensure all the requirements clauses are covered" when answering demographic questions about different documents or models. This suggests that the format or standard that a document follows affects the completeness of requirements for a software system. This is important when the receiver is concerned with the completeness of information. The second theme is related to the completeness of the ReqSpec document. One participant mentioned that he or she ran out of time answering questions possibly because of some incomplete sections in the document. This suggests that there was a relevance problem concerning insufficient information.

Another theme we find is related to the accuracy of information. Two participants mentioned problems in mismatch of code and documentation as well as incorrect specification from stakeholders. As well, one participant reported that the UCM document might have an error when answering P1Q7. Although there were only a few responses regarding accuracy of information, this theme of questioning the accuracy of information is important to communication. As one of the participants commented, "incorrect specs from client leads to incorrect implementation" which leads to waste of time and bad product. Accuracy of information is also one of the important attributes of communication in Chapter 5.

In the experiment, up-to-date information is important to participants. However, it seems difficult to keep information up-to-date as commented by five participants. Participants as receivers might suspect the information, as in "is the information up-to-date?", "what information has changed?". Similarly, traceability of software artefacts seemed to be important to two participants. Traceability could also be one of the questions that the receivers have about software artefacts.

The last theme we find is related to the manageability of stakeholders. This theme also appeared in Chapter 5 as one of the communication problems. Although the survey did not show if this problem is significant in communication among stakeholders, we find one of the participants mentioned that manageability of stakeholders was a problem in his or her organisation. The participant commented that there were too many stakeholders to deal with. Stakeholders sometimes sent different information. This in turn caused confusion for developers and thus a waste of time.

6.4 Threats to Validity

In this section we discuss the threats to the validity of the experiment and the mitigations to these threats. We assess the threats to validity based on Figure 6.39, which shows the cause-effect construct for the experiment. The figure is based on the validity threats discussion by Trochim [113]. In the experiment, we theorise that the use of requirements documents with improved transparency (a cause) can lead to a more effective communication (an effect). This is the top part of Figure 6.39. We then operationalise our theory to the bottom part of Figure 6.39. We evaluate the requirements documents with accessibility, understandability, and relevance as discussed in Section 6.1.5. We then observe how well our participants answer questions using the ReqSpec document and the UCM document. In the following sections, we first discuss the conclusion validity and internal validity that are related to the observation part of the experiment. We also discuss the construct validity which concerns the operationalisation of theory to observation. Finally, we discuss the external validity of the experiment.

6.4.1 Conclusion Validity

As illustrated in Figure 6.39, conclusion validity concerns the conclusions made about the program-outcome relationship. In the experiment, we need to be aware of any threats that could affect the conclusions made about our hypotheses, EH1 (time), EH2 (correctness), and EH3 (confidence). There were several potential threats in the experiment that could lead us to incorrect conclusions.

Similar to the threat to conclusion validity of our survey, the main threat in the

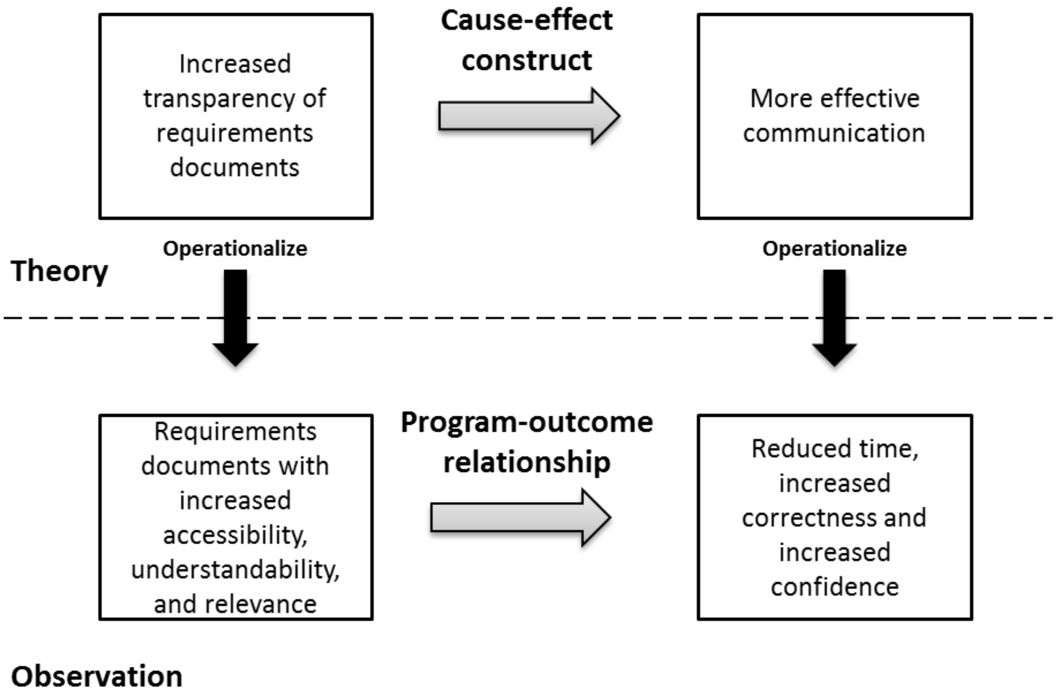


Figure 6.39: Cause-effect construct for our experiment.

experiment was the conversion of ordinal scale (Very good, Good, Satisfactory, Poor, Very poor) to numerical values. In the analysis of our experiment, we transformed the Likert scale for P2Q6b into numerical values for testing EH3. We also transformed the Likert scale for P2Q5 and P2Q6a for analysing participants' assessment on the accessibility, understandability, and relevance of information. We then performed t-tests on the Likert data to see if there were any statistically significant differences. The use of parametric statistical tests on Likert data could violate statistical rules. This in turn could affect our conclusions about the differences between the UCM document and the ReqSpec document. However, according to Norman [81], parametric statistical tests could be used for analysing Likert data without the "fear of coming to the wrong conclusion". Therefore, it was reasonable to use parametric tests for analysing our data.

In our analysis, we used two-tailed tests to test the three hypotheses, EH1, EH2, and EH3. The two-tailed tests helped us to determine if there were any statistically significant differences between the two treatments regardless of the directions of such differences. To test if the UCM treatment is significantly better than the ReqSpec treatment, one-tailed tests could be used. Additional hypotheses for our experiment could be constructed to capture the direction of interest. For example, we could have a new hypothesis in addition to EH1: the time spent by participants using the UCM document is less than the time spent by participants using the ReqSpec document to answer questions in Part 1 of the questionnaire. In this thesis, we determined if there were any differences between the two treatments by using two-tailed tests. For future analysis, one-tailed tests would be appropriate for detecting effects in the direction of interest.

In our experiment, how we assign the requirements documents could also threaten the conclusion validity. If our assignment was biased toward one of the treatments, for example participants with more experience received the UCM document, the conclusion of the experiment could favour that treatment. To minimise assignment bias, we randomly assigned documents to our participants in the experiment. This made sure that the assignment of the documents was not bias toward the UCM document, which was more transparent than the ReqSpec document.

We also have another threat regarding the random heterogeneity of participants in our experiment. According to Wohlin et al. [122], individual differences could have a larger effect on the outcome of the experiment than the treatments if subjects were very heterogeneous. In our experiment, we focused on two types of participants, students and software professionals. The year of study or the amount of work experience that our participants have could affect how they use the requirements documents. However, from the statistical analysis, we did not find the difference in the experience of our participants has a significant influence on the effectiveness of the documents.

6.4.2 Internal Validity

Internal validity is concerned with whether there is a causal relationship between the program and the outcome. When conducting our experiment, we need to ensure that the outcome of the experiment is caused by the two treatments and not caused by uncontrolled or unknown factors. There were several potential threats to the internal validity of our experiment.

The first threat was related to participants' reactions during the experiment. Since the experiment involved participants' reading requirements documents and answering the questionnaire, our participants could feel frustrated if they could not find answers. They would also feel tired if the questionnaire took too long to answer. This might lead participants to answer randomly or not to answer the questionnaire at all. To reduce the likelihood of participants' feeling negative about the experiment, we tried to minimise the length of the questionnaire and to include questions that could be answered easily.

Another threat to the internal validity was how we selected our participants. We used convenience sampling, in which our participants volunteered to take part in the experiment. Our participants might be more motivated than the population in the software industry. Our participants might feel more positive about the requirements documents than the population in the software industry. This in turn could affect the results of how helpful the requirements documents were in answering questions. However, how our participants perceive the requirements documents did not have a significant impact to the hypothesis testing. This was because the goal of the experiment was to compare the effectiveness of the requirements documents. The focus of the analysis was on determining which requirements document was more effective in helping participants to answer questions.

6.4.3 Construct Validity

Construct validity is concerned with the operationalisation links between theory and observation of the experiment. In our experiment, we needed to evaluate whether the questions constructed actually tested the three hypotheses (the second operationalisation link from the effect to the outcome). The main threat to the construct validity was the experimental design. We used the questionnaire as a set of questions that a receiver has in the communication model for our participants. We also used the questionnaire to collect participants' opinions about the requirements documents.

The wording of the questions could be a threat to construct validity, which would also affect the conclusion validity. If the questions were ambiguous, our participants might be unable to answer them. Participants could also misunderstand the questions and give us wrong answers. In the experiment, we found that question P1Q4 might be ambiguous to our participants using the UCM document. There was a confusion between the word "requirements" used in the question and the word "prerequisites" used in the UCM document. However, we did not find any significant impact on the correctness of answers collected for that particular question.

The length of the questionnaire could also be a threat. Some questions might be left unanswered as we limited the time that our participants could spend in the experiment. Our assessment on the effectiveness of the requirements documents might be affected by incomplete answers. To mitigate this threat, we minimised the number of questions in the questionnaire. Participants were asked not to spend too much time on each question. They were also asked to note down any problems if they could not answer the questions. In addition, we conducted a pre-test to see how long it took to read the requirements documents and to answer the questionnaire. This pre-test helped us to improve the format of the questionnaire.

We also constructed the questions based on the wording used in the ReqSpec document. Participants could identify keywords from the questions and thus find information in the requirements documents. This could help participants to answer the questions within the time of the experiment. In addition, we tried to minimise bias in favour of the UCM document by constructing the questionnaire based on the ReqSpec document. For example, we asked participants about the requirements for handling hard-copy applications in P1Q4. The question was constructed based on the wording in the ReqSpec document.

The difference in the year of study or work experience might affect the outcome of our experiment which in turn would affect the conclusion of our experiment. For example, graduate students might perform better in the experiment than undergraduate students because they were more likely to have more experience with different software artefacts. However, the data collected from the experiment did not show any significant difference in how well the participants answered the questionnaire with respect to their experience.

Another threat to the construct validity is related to how the participants assess their confidence in understanding the requirements documents. The participants might try to give a better score about how well they understood the documents and how well they answered the questionnaire. Their self-assessments might not represent what they actually thought. This threat could affect the assessment of which requirements document was more effective in helping participants to answer questions. To determine the effectiveness of the documents, we not only looked at participants' self-assessments but we also looked at the time spent and the number of questions answered correctly by participants in the experiment.

How we assessed the correctness of answers to Part 1 of the questionnaire was a potential threat to construct validity. The correctness of the answers was determined by the researcher. The researcher might misinterpret the answers and which consequently affected the conclusions made about the effectiveness of the requirements documents. It was not easy to evaluate answers objectively as the questions in Part 1 of the questionnaire were open-ended. To minimise the risk of misinterpretation, we also looked at the locations where they found the information to answer some questions in Part 1. This helped us to ensure that participants looked in the right locations for the answers rather than guessing the answers. If the participants guessed the answers, this could affect our analysis about the degree of transparency of documents. This in turn could also lead us to wrong conclusions about the documents. In the experiment, we found a few participants inferred the answers from screenshots instead of from the actual text.

In the experiment, we believe that the UCM document was more transparent than the ReqSpec document. One potential threat to construct validity was that the UCM document was not more transparent than the ReqSpec document. The effectiveness of the documents in presenting functional requirements to participants could be caused by other constructs such as the experience of participants. Such threat could affect our conclusions about transparency and its usefulness in requirements engineering. However, we analysed the degree of transparency in the documents based on the three attributes of transparency in Section 6.1.5. The preliminary transparency analysis showed that the UCM document was more accessible and relevant than the ReqSpec document. Moreover, we found the UCM document was more accessible and relevant to our participants than the ReqSpec document from participants' responses to the questionnaire. Furthermore, we did not find any significant differences in the effectiveness of the documents with respect to the experience of participants. Therefore, based on the preliminary transparency analysis and the results of the experiment, the UCM document was more transparent than the ReqSpec document.

6.4.4 External Validity

External validity concerns the generalisation of the theory on top of Figure 6.39. The main threat to external validity concerns our sampling method for the experiment. We can not generalise our results to the population in the software industry as our participants are mainly students and a small number of software professionals. It is difficult to determine how well the knowledge and experience of the student participants have in comparison with the population in the software industry. Our results are also limited to one aspect of requirements engineering as we focus on presenting functional requirements of a software system. However, the data collected from the experiment helps us to compare the effectiveness of the two requirements documents. The results also give us high confidence to support our hypotheses.

6.5 Summary

In this chapter we presented the design and the execution of our experiment. The experiment was conducted to evaluate how useful transparency was in presenting functional requirements of a software system using two types of requirements documents. We determined that the UCM document was more transparent to our participants than the ReqSpec document for answering questions about the functionality of a software system. In Section 6.3, we presented the descriptive statistics of the experiment as well as the hypothesis testing for EH1, EH2, and EH3. We also organised the comments made by our participants into different themes.

The statistical analysis gives us high confidence to support our hypotheses. Firstly, we find that there is a difference in the time spent to answer questions between participants using the ReqSpec document and participants using the UCM document. On average, participants who used the UCM document spent less time answering questions in Part 1 of the questionnaire. Secondly, there is a difference in the number of questions answered correctly by our participants. The results show that, out of four questions, participants using the UCM document answered more questions correctly with an average score of 3.34 than participants using the ReqSpec document with an average score of 2.48. We also find that participants who used the UCM document were more confident about their answers than participants who used the ReqSpec document. The results suggest that the UCM document was more effective than the ReqSpec document for participants in answering questions about the functional requirements of a software system.

In the experiment, we observe the experiential differences between Undergraduate participants and Graduate + Industry participants. The statistical analysis shows no significant difference between the two groups of participants in the time spent in answering questions and the number of correct answers. The analysis also shows that there is no difference between the two groups of participants in the confidence of the correctness of their answers.

The UCM document as we determined was more transparent than the ReqSpec document. We find that the UCM document was more accessible to our participants for identifying locations of information than the ReqSpec document. The results also suggest that fewer participants went through different parts of the UCM document than those using the ReqSpec document to find relevant information. Moreover, participants tended to be more satisfied with the UCM document than those using the ReqSpec document. The findings from the experiment show that transparency is a useful attribute in presenting functional requirements of a software system.

In the qualitative analysis, we find several themes that affect the assessment of transparency of the communication channel. The themes relate to the assumptions about transparency in Chapter 3. We also find themes that affect the accessibility, understandability, and relevance of the channel. In addition, we find other interesting themes such as the need for up-to-date information that emerge from participants' comments.

In the following chapter, we discuss the research questions that we addressed in this thesis. We also discuss interesting points from the exploration and evaluation of transparency. In addition, we discuss limitations of our research and improvements on our survey and experiment.

7

Discussion

In this thesis, we have explicated the concept of transparency for software engineering by exploring existing notions of transparency and introducing a working definition of transparency for software engineering. The exploration stage helped us to define a clear picture of the concept of transparency and its boundaries for software engineering. We used the working definition to observe and describe how transparency in requirements documents affected stakeholders' ability to answer questions. Our evaluation of the importance of transparency in software engineering involved the use of two types of requirements documents with different degrees of transparency in an experiment. The purpose of the experiment was to support the hypothesis about the usefulness of the concept of transparency in software engineering.

The exploration and evaluation of transparency have brought to light some interesting points. These points help us to identify areas for future investigation of transparency in software engineering. In this chapter, we discuss these points and inferences from our research findings. We also discuss limitations of our research and improvements on our survey and experiment. Before we discuss the interesting points from our research, we first revisit our research objectives in the following section. We discuss the research questions that we addressed in this thesis.

7.1 Revisiting the Research Objectives

In this thesis, we explored two questions: what is transparency in software engineering and how useful is transparency to software development. This thesis has two stages, exploration and evaluation. In the exploration stage, we aimed to gain insights into the basic concept of transparency and its relation to software engineering. We answered the following two research questions:

- RQ1. How much does the term "transparency" occur in the software engineering literature?

  The term "transparency" is widely used in the software engineering literature. In the literature search, the term "transparency" appears in different areas of software engineering. In Chapter 3, we look at how transparency is defined and used in the following areas of software engineering: information privacy; computer ethics; security, trust, and risk management; visual notations; agile development; dependable systems; and requirements engineering. A percentage count or a systematic literature review on how much the term "transparency" occurs in the literature might fully answer RQ1. However, the main purpose of the exploration stage is not to do a precise count on the occurrence of the term "transparency" in software engineering. The main purpose is to understand the concept of transparency and its relation to software engineering. The literature search reveals the use of transparency in different areas and the diversity of the implications of transparency in software engineering. For example, transparency implies the visibility of information about a software project in agile development. In the context of visual notations or graphical representations, transparency implies the meaning of information could be inferred from its appearance.

- RQ2. What is the concept of transparency in the software engineering context?

  We define transparency in software engineering as *the degree to which stakeholders can answer their questions by using the information they obtain about a software system during its life cycle.* This definition of transparency is based on the implications found in Chapter 2. We evaluate our preliminary definition of transparency by using a survey (Chapter 5). The survey results suggest that our definition is important for communication in software projects. Moreover, this definition of transparency rests on three attributes: accessibility, understandability, and relevance. These attributes affect stakeholders' ability to see the information to achieve their goals. The survey results also suggest that the three attributes are important or very important to our definition of transparency.

At the evaluation stage, we answered the third research question:

- RQ3. How important is the concept of transparency to successful software development?

  To answer RQ3, we construct a set of hypotheses that show the importance of transparency to different aspects of software development. We illustrate the importance of transparency in requirements engineering through an experiment (Chapter 6) based on responses to a questionnaire from software professionals and students. The experiment demonstrates the usefulness of transparency in presenting functional requirements of a software system to software developers and tertiary students. The results from the experiment imply that the concept of transparency can be useful for improving the software development process.

We have addressed the three research questions by collecting evidence about transparency from a literature review, an exploratory survey, and a controlled experiment. In the following section, we discuss the evidence collected from the survey and the experiment.

7.2 Exploring Transparency

In the exploration stage, we explored existing notions of transparency from different areas including software engineering (Chapter 2 and Chapter 3). We also collected opinions from stakeholders, mainly software developers, about communication problems in software projects and transparency in different contexts (Chapter 5). The exploration revealed different implications of transparency used in different areas. It also revealed the lack of specific measurable characteristics and ways to measure the concept of transparency in software engineering. We discovered two interesting points from the exploration.

The first point relates to the communication problems reported by our participants in the survey. When our participants were receivers of information, they reported more problems with inaccessible and non-understandable information in a software project. As senders of information, they have problems with stakeholders understanding their information more frequently than other types of problems. This type of asymmetry in the social sciences indicates an “actor-observer” bias or a “superiority bias”. In an actor-observer bias, the observer of an action is more likely than the actor of that action to ascribe fault to the actor [64]. In a superiority bias, members of any group are more likely to judge outsiders more harshly than other members within their own group [49].

The two types of bias could explain the asymmetries found in our survey. Our participants could suffer from an actor-observer bias as they tended to report the communication problems being the fault on the other side of the communication model. Our participants might also suffer from a superiority bias as they could consider themselves to be above-average in communication skills and the other side of the communication model being below-average in communication skills. Therefore, the bias can affect the sender or receiver's assessment of the degree of transparency of information during communication in the software life cycle. This in turn can affect senders' ability to accurately articulate communication problems.

The second point also relates to the communication problems reported by our participants in the survey. Our participants have reported inaccurate information as another common problem in software projects. Problems with the accuracy of information can affect communication in software projects as discussed in Section 1.4. They may also affect receivers' assessment of the degree of transparency of information presented in the communication channel. When the information is inaccurate, receivers may judge the information as irrelevant or insufficient to answer their questions. Receivers may thus assess the inaccurate information as non-transparent. Therefore, the accuracy of information has some effect on the receivers' ability to answer their questions. The effect of the accuracy of information will depend on the type of receivers' questions. It seems that accuracy could be the fourth attribute of transparency as it affects the receivers' ability to answer questions. However, we believe that accuracy is not an attribute of transparency. This is because inaccurate information could still be accessible and understandable to receivers. The relevance of inaccurate information would depend on what questions receivers have. If accuracy of information is not a concern for receivers, inaccurate information is transparent to receivers when it is accessible, understandable and relevant to their questions. Hence, accuracy is an attribute of information that enables or hinders transparency.

Similarly, other attributes of information such as valid and up-to-date information as mentioned by our participants may also affect receivers' ability to answer their questions. How much these attributes enable or hinder transparency will also depend on the questions that receivers have. For example, if a receiver is concerned with the latest requirement changes, the only relevant information to the receiver is about the requirements being up-to-date. The second point suggests a need to investigate how different attributes of information may affect receivers' ability to answer their questions. There is also a need to analyse different types of questions a receiver may have during communication in the software life cycle.

7.3 Evaluating the Importance of Transparency

In the evaluation stage, we designed and conducted an experiment to compare two different types of requirements documents in presenting functional requirements of a software system. The results of the experiment (Chapter 6) demonstrated that a more transparent requirements document was more effective for software developers and tertiary students to answer questions about the functional requirements of a software system than a less transparent requirements document. We discovered three interesting points from the evaluation.

The first interesting point relates to the understandability of the requirements documents. From the comments made by our participants in the experiment, most of them could understand the requirements documents. According to our participants who used the UCM document, the use case diagram was useful in helping participants to understand the functionality of the software system. However, a few of our participants commented that the use case diagram was insufficient in the UCM document. They suggested that more diagrams such as workflow diagrams would improve understandability. Interestingly, participants who used the ReqSpec document suggested that the ReqSpec document should include use case diagrams and other diagrams such as sequence diagrams to help readers understand the software system. In addition, participants in both treatments suggested that using pictures or illustrations could help readers to understand the information presented in the requirements documents.

It seems that diagrams or pictures can help receivers understand information. The number of diagrams or pictures presented in software artefacts can affect receiver's ability to understand information. In the experiment, more diagrams or pictures may have improved understandability of information in the requirements documents. However, this is not always true. If our participants were unfamiliar with the notations used in the requirements documents, problems such as misunderstanding of requirements as discussed by Al-Rawas and Easterbrook [2] can occur. This raises a question for future research: How much do diagrams or pictures affect the transparency of information presented in software artefacts?

The second interesting point relates to the relevance of the requirements documents. In the experiment, problems relating to unclear or missing detailed information affected our participants' ability to find information. For example, 23 of 58 participants (approximately 40%) commented that they could not find relevant information to answer P1Q7 using either the ReqSpec document or the UCM document. P1Q7 was a tricky question to our participants in the questionnaire as neither document had clear, specific information. Some participants also could not find relevant information at the expected location to answer P1Q7 and tried to look for relevant information in other locations of the documents again. It seems that missing or incomplete information in the documents affects the assessment of relevance. This raises another question for future research: How does missing or incomplete information affect the transparency of information presented in software artefacts? This point is also related to the point discussed previously regarding how different attributes of information may affect receivers' ability to answer their questions.

In addition to missing or incomplete information, other interesting themes arise from the comments on concerns about software artefacts or communication in general. Two of the concerns focus on accuracy of information and the need for up-to-date information. These two concerns also appeared in the point discussed previously regarding the effect of different attributes of information on receivers' ability to answer their questions.

Lastly, we find an interesting observation from the experiment: the participants who used the ReqSpec document might have inferred the answer to P1Q6 from information such as screenshots instead of from the actual text. This might affect the assessment of transparency. The inference of information might give participants a false sense that the information presented in the ReqSpec document is adequate to answer their questions. Participants might interpret the meaning of information differently to what is originally intended and thus think that they have answered their questions sufficiently. This suggests a need to consider how receivers use information when building a diagnostic framework, also future work for research in transparency in software engineering.

7.4 Inferences

In this section, we discuss the inferences drawn from our research findings. We first discuss the application of transparency in software engineering. We then discuss our working definition of transparency in the context of the three attributes of transparency. We discuss attributes of information that may affect receivers' assessment of transparency.

7.4.1 Application of Transparency

The results from the experiment demonstrate transparency's usefulness to requirements engineering. The results show that a more transparent requirements document tends to be more effective than a less transparent requirements document in presenting information to software developers and tertiary students. The findings from the experiment suggest that having a transparent requirements document is useful for conveying requirements to stakeholders who have some knowledge or some experience in requirements engineering.

Stakeholders can understand the functionality of a software system in a shorter period of time using a transparent requirements document. They can be more confident about the information obtained from a transparent requirements document. The findings from the experiment also suggest that the degree of transparency of a requirements document depends on how a receiver uses and perceives such document. The presentation of information in a requirements document affects how a receiver uses the document, which in turn affects how a receiver assesses the accessibility, understandability, and relevance of information.

Transparency, as illustrated from the findings of the experiment, can affect how well a communication channel conveys information to receivers. Hence, increasing the degree of transparency in software artefacts improves stakeholders' ability to answer questions about a software system in development. Good transparency enables stakeholders to obtain information within a reasonable amount of time, and stakeholders are likely to be more confident about the information. For example, software project managers can be more confident about the project status when software artefacts that contain information about their software projects are transparent. Then, they ask fewer questions about the software developers and their work for the project when the information provided by software developers about what they are doing is transparent. A transparent software artefact can thus reduce the number of stakeholders' questions and can lead to a more effective assessment of the software system. It also reduces the number of communication channels needed for the sender to convey information to different stakeholders. This in turn improves the software development process. Similarly, good transparency improves the information presented about a software product. For example, users have a better understanding of the features provided in the software product consulting transparent user manuals.

To improve the software development process, or to improve the information about a software product, our preliminary set of questions for determining the degree of transparency is useful. As illustrated in the experiment, the questions in Table 6.1 (Chapter 6) are applicalbe to analyse transparency of a software artefact. In the experiment, these questions determined the accessibility, understandability, and relevance of the requirements documents. This set of questions is the starting point for building a diagnostic framework. It helps stakeholders at the sender side to diagnose problems in the communication channel during the software life cycle. For example, according to the interviews conducted by Al-Rawas and Easterbrook [2], one programmer complained that he had to read a large amount of text to understand a single requirement. If the sender of requirements (a requirements engineer or a software developer) had our set of questions, he would diagnose that the information had problems with relevance. The programmer encountered relevance problems where he could not answer his questions within a reasonable amount of time and the information was not directly connected to his questions. Therefore, the set of questions for determining transparency is useful to software developers. In addition, software developers can use this set of questions to explicitly think about the concept of transparency when communicating with other stakeholders.

7.4.2 Attributes of Transparency

We have conceptualised transparency in software engineering and have argued that stakeholders need communication with three attributes of transparency to answer their questions. Firstly, accessibility is important as it concerns how easily stakeholders can access information to answer their questions. Accessibility must come before understandability and relevance of information. Next, stakeholders need understandable information to answer their questions. Lastly, if the information is relevant to their questions, stakeholders can answer their questions within a reasonable amount of time. Therefore, these three attributes are important for enabling stakeholders to use the information to answer their questions.

We believe that our working definition is useful in software engineering as it reduces the ambiguity of existing notions of transparency. For example, Dabbish et al. [30] stress that transparency supports communication and coordination behaviours in software development as it makes work visible to stakeholders. According to Dabbish et al., transparent development environments allow everyone to "see and have meaningful access to (almost) everything". However, from the literature, it is unclear how stakeholders see information or what a meaningful access is. Our definition is clear about how well stakeholders "see and have meaningful access" to information. Stakeholders should be able to answer their questions using the information created in transparent development environments. Accessibility and understandability determine how well stakeholders see information. Stakeholders can see information if they can access and understand the information. We can also determine how meaningful the information is to stakeholders with relevance. The information is meaningful to stakeholders if it is relevant to answering their questions. Therefore, our working definition is useful in many ways for improving existing notions of transparency.

However, our working definition is restricted. This is because we did not specify the behaviour of the sender of information or the truthfulness of information provided to stakeholders. Our definition depends on the judgement of stakeholders who obtain the information during the software life cycle. As discovered in the exploration and evaluation stages of our research, other attributes of information may affect the assessment of transparency. Some existing notions of transparency as we explored in software engineering also imply that transparency has other attributes such as accuracy. For example, Ghezzi et al. [48] assert that transparency makes a development available and easily accessible for examination. They further suggest that a software product is visible if "it is clearly structured as a collection of modules, with clearly understandable functions and available and accurate documentation". This implies that a transparent software product has the following attributes: understandability, availability, and accuracy of information.

Another example of different attributes of transparency is in the discussion by Sampaio do Prado Leite and Cappelli [95]. Sampaio do Prado Leite and Cappelli propose that transparency is a general quality or a non-functional requirement for a software system which relates to information disclosure. They identify 33 quality attributes that contribute to transparency. Attributes such as accuracy and completeness are among the 33. Two of our attributes of transparency - accessibility and understandability - are also related to their definitions for accessibility, informativeness, and understandability. However, as discussed in Section 3.2.7, our third attribute of transparency - relevance - is not in their list of 33 attributes. Moreover, their definitions of the quality attributes are ambiguous, and the context is unclear. Many of their quality attributes seem to depend on the type of questions that stakeholders have. Furthermore, it is unclear how each of the 33 quality attributes affects stakeholders' ability to answer their questions. It is also unclear how these attributes affect the quality of information being disclosed to stakeholders. An investigation of how other attributes of information may affect the assessment of transparency can be carried out in future.

Although Sampaio do Prado Leite and Cappelli provide a list of attributes that contribute to transparency, it seems difficult to apply all 33 attributes to achieve transparent communication in software engineering. It also seems difficult to analyse the degree of transparency in software engineering using all 33 attributes. Moreover, many of their quality attributes are dependent on the context in which stakeholders are situated. Not all of their quality attributes contribute to transparency in software engineering every time, whereas our three attributes are important to achieve transparency in software engineering. As an example, one of their quality attributes - operability - is not applicable to our experiment, because "the quality of being treated by surgical operation" is not a concern for the requirements documents. Our three attributes - accessibility, understandability, and relevance - can be more easily applied for analysing the degree of transparency in software engineering than Sampaio do Prado Leite and Cappelli's list of the 33 attributes. The time it takes to analyse transparency would be shorter using our attributes than using Sampaio do Prado Leite and Cappelli's attributes.

In summary, our working definition of transparency is useful to clarify ambiguous existing notions of transparency in software engineering. The three attributes of transparency are useful in determining how well stakeholders answer their questions about a software system during its life cycle. The next step is to formalise our working definition of transparency in software engineering. It is important to establish a formal definition to help researchers to properly observe and interpret transparency in software engineering. The formal definition will also help researchers to consolidate ideas relating to transparency in software engineering. In addition, it will be important for constructing a credible diagnostic framework based on an authoritative definition of transparency. Collecting supporting evidence will formalise the definition of transparency. Evidence such as the validity and appropriateness of the definition in software engineering will be required.

In this thesis, we have conducted two empirical studies (survey and experiment) which supported our claims about transparency and its usefulness in software engineering. We have also demonstrated the evaluation of transparency using our working definition in the preliminary transparency analysis of the two requirements documents for the experiment. However, the evidence that we collected in this thesis is limited. For example, the results of the experiment demonstrated the usefulness of transparency in one aspect of software engineering (requirements engineering). To formalise the definition in software engineering, evidence from different aspects of software engineering is needed. In the following section, we describe the limitations of our research.

7.5 Limitations

We have presented the results from a survey in Chapter 5, in which we explored communication problems from a software project stakeholders' point of view. The responses collected from the survey showed that a majority of our participants frequently or always encountered transparency problems in software projects. The responses also showed that a majority of our participants were familiar with the term "transparency" used in more than one context.

The main limitation of our survey was that we could not generalise the results to the entire population involved in the software industry. However, the findings from the survey helped us to gain insights into how frequently different types of communication problems might occur in a software project. It also gave us an indication of the types of communication problems software developers might encounter in a software project. Furthermore, we identified other attributes such as accuracy that might be important to communication in a software project. The attributes might also enable or hinder transparency in software engineering.

In the evaluation of transparency, we have conducted a controlled experiment as presented in Chapter 6. The experiment demonstrated the usefulness of transparency by comparing two types of requirements documents with different degrees of transparency. The results of the experiment showed that a more transparent document was more effective than a less transparent document in terms of the time spent by participants, the correctness of participants' answers, and participants' confidence level in their answers.

There were three limitations of our experiment. Firstly, our experiment was limited in testing one aspect of software engineering, only requirements engineering. The findings from the experiment showed that the UCM document was useful for presenting functional requirements of a software system. Changing the questions could change the degree of transparency of the requirements documents. This is because our definition of transparency depended on the stakeholders' ability to answer their questions using the received information. For example, if we asked participants to find non-functional requirements in the UCM document, the UCM document could become irrelevant to participants as there was no information on non-functional requirements. Similarly, if we provide a different software artefact such as a software architecture document to our participants rather than a requirements document, the software artefact might not be transparent at all. Our participants might have problems understanding the information presented in the software artefact. They might also have problems with relevance of information as the software artefact might contain no information about requirements of the software system.

Secondly, our experiment was limited in testing whether participants have really learned what the software system could do. In the experiment, participants were asked to answer questions directly while reading requirements documents. The aim of the experiment was to test how well the requirements documents were in helping participants to understand the information presented within a limited amount of time. The focus of the experiment was not on whether participants learned what they needed to know after reading the requirements documents.

Lastly, our experiment was limited with respect to the type of participants we recruited. Since the requirements documents were specific to software engineering, the target audience was limited to stakeholders who have some knowledge about software development or experience in the software industry. They also have some experience using different software artefacts. If non-expert stakeholders such as end users were involved in our experiment, the results might be different. For example, the UCM document might be less useful to end users than the ReqSpec document because end users might not know any notations for use case diagrams.

Although there were limitations in our survey and experiment, we have made progress towards a better definition of transparency in software engineering. We have also collected evidence to support our hypotheses about the usefulness of transparency in software engineering. In the following section, we discuss how the survey and the experiment can be improved for any researchers who wish to reuse our survey questionnaire or to reproduce our experiment.

7.6 Improvements on Survey and Experiment

During the course of the survey, we isolated three improvements that might be needed for anyone who wished to reuse the questionnaire. Firstly, the demographic questions could be improved to include more details about the participants. Our demographic questions relied on participants' self-assessments for their knowledge and experience in software engineering. It was difficult to divide the participants into distinctive groups from the responses collected. This in turn made it difficult to observe the differences in the types of communication problems encountered by different groups of participants. To improve the survey, questions about participants' years of work experience and the types of software projects that they were involved in could also be included.

The second improvement was to stress the importance of answering questions in the survey. We discovered that there were a few missing or incomplete answers from the responses collected. The results could be affected significantly if we had a large number of incomplete responses. Therefore, questions where responses were important to address the research questions could be emphasised in the survey to the participants.

Lastly, another improvement on the survey is related to the wording of the questions. Some questions were broad and ambiguous such as the wording of Q7 and Q12 in the questionnaire as discussed in section 5.4.3. To be more specific about communication problems and transparency in software engineering, we could describe different scenarios in the questionnaire. This would help participants to focus on particular areas in software engineering.

For administering the experiment, we suggest two improvements to reproduce the experiment. Firstly, the experiment was limited in measuring how well people understood the information presented in the requirements documents. We could not assess if participants actually understood the information based on their responses. Participants could copy answers directly from the documents without understanding the content. To improve this, another session which involves participants in building a prototype of the system to test their understanding could be held.

Secondly, we found some participants did not read the instructions before starting the experiment. We did not require participants to read through the entire documents, but some did. Moreover, a few of our participants noted down the start time after finishing reading the documents. This could affect the time spent by our participants. However, we did not notice any significant impact on the results from the analysis. To improve the experiment, any ambiguous instructions should be clarified before commencing the experiment. The experiment could also be divided into three parts with one part of the experiment given at a time.

7.7 Summary

In this chapter, we discussed the answers to our research questions. Discussion included interesting points from the exploration and evaluation of the concept of transparency in software engineering. The interesting points were related to communication problems in software projects, and factors as well as different attributes that could affect the assessment of transparency in software engineering. We also discussed inferences such as the usefulness of transparency to improve the software development process from our research findings. In addition, we discussed the limitations of our research and improvements for the survey and the experiment. For example, the wording of survey questions and the procedure for the experiment could be improved to minimise ambiguity.

In the next chapter, we summarise the thesis and contributions of our research. We also suggest areas for future research and conclude with thoughts about our research.

8 Conclusion

8.1 Summary

The term “transparency” is widely used in software engineering as an important concept. It appears in much of the software engineering literature with different implications but without a proper definition of what it actually means. The term “transparency” implies the notion of information being visible or open to stakeholders. Paradoxically, it also implies that the information is not easily seen or noticeable to stakeholders. These implications of transparency are useful to various aspects of software engineering. In particular, transparency’s implication as visible information is useful to improve communication among stakeholders during software development. A lack of transparency hinders communication in software development. However, there is very little investigation of how transparency’s implication of visible information might help software development. Moreover, it is unclear what transparency is improving or how transparency might be assessed in software engineering. These questions motivate this researcher to explore the term “transparency” with its implication as visible information in software engineering. The concept of transparency refers to the implication of making information visible to stakeholders throughout the thesis.

In this thesis, we argue that the concept of transparency is important in software engineering because it is important for stakeholders to easily see information during communication in software development. The concept of transparency will benefit software developers in communicating information to stakeholders during the software life cycle. Since the term is not clearly defined in software engineering, we need to first explore notions of "transparency" from different areas. The exploration is the fundamental stage that defines a clear picture of transparency and its boundaries in software engineering. We evaluate the usefulness of the concept of transparency after the exploration. Therefore, this thesis involves the exploration and evaluation of the concept of transparency in software engineering. In Chapter 4, we discuss the scope of our research approach and set hypotheses for research question 3 (RQ3).

To gain insights into what transparency should be and how it relates to software engineering, a literature review on the concept of transparency from different areas is important. In Chapter 2, we explore how the concept of transparency is defined in philosophy, organisations, business ethics, public participation, and computing. This exploration reveals three common attributes of transparency useful in software engineering. These attributes concern the accessibility, understandability, and relevance of information. The degree of transparency of information depends on the sender who controls the information sent to the receiver. It also depends on the time, the means, and receiver's skill to communicate with the sender.

In Chapter 3, we propose a working definition for transparency in software engineering based on the common attributes from Chapter 2. We define transparency as *the degree to which stakeholders can answer their questions by using the information they obtain about a software system during its life cycle.* This definition is tentative as a starting place to help us to observe and interpret transparency's usefulness in software engineering. We assert that accessibility, understandability, and relevance are important attributes of transparency. These attributes are important for enabling stakeholders to answer their questions about a software system.

In Chapter 3, we review existing notions of transparency in the following software engineering-related areas where the notion of transparency concerns visibility or openness of information: information privacy; computer ethics; security, trust, and risk management; visual notations; agile development; dependable systems; and requirements engineering. The literature review of transparency in software engineering is important for helping us to see how transparency was defined in existing literature. It is also important for us to see how our definition related to existing notions of transparency in software engineering. The literature review reveals a lack of specific characteristics in current definitions and ways to explicitly assess transparency.

In addition to the literature review, we conduct a survey to collect evidence to answer the research question: what is transparency in the software engineering context? The survey results in Chapter 5 reveal that a majority of our participants frequently or always encounter transparency problems in software projects. The results also reveal that a majority of our participants are familiar with the term “transparency” used in more than one context. Moreover, our participants indicate our definition of transparency is important to communication in software projects. The three attributes are rated as important or very important to our definition of transparency. The evidence collected from the survey gives us confidence about our definition of transparency and its relation with communication in software projects.

At the evaluation stage of transparency, we gather evidence to support our claim about the usefulness of transparency in software engineering. In Chapter 6, results of an experiment illustrate the importance of transparency in requirements engineering. The experiment involves the comparison of the effectiveness of two requirements documents, ReqSpec and UCM, with different degrees of transparency. From the experimental analysis, we conclude that the UCM document is more effective in presenting the functionality of a software system than the ReqSpec document because the UCM document is more transparent than the ReqSpec document. The analysis also reveals that the previous software experience of our participants does not have a significant impact on participants’ ability to use the requirements documents to answer questions.

In Chapter 6, we also illustrate a preliminary analysis of transparency of the two requirements documents using a set of questions for determining the accessibility, understandability, and relevance of information. Based on the preliminary analysis of transparency, we determine that the UCM document is more transparent than the ReqSpec document by our definition. The findings from the analysis of the effectiveness of the requirements documents thus support our hypothesis about a transparent requirements document being more effective for software developers to answer questions about the requirements of a software system than a non-transparent requirements document. The results of the experiment demonstrate the usefulness of transparency in presenting functional requirements of a software system to software developers and tertiary students. The experiment helps us to gain confidence about the value of transparency in one aspect of software engineering, requirements engineering.

In summary, this thesis explores the concept of transparency and demonstrates transparency’s usefulness in software engineering. Our findings suggest that the degree of transparency affects how well a communication channel conveys information to the receiver. Increasing the degree of transparency in software artefacts would improve stakeholders’ ability to answer questions about a software system. This implies that stakeholders should obtain information within a reasonable amount of time and be more confident about the information with a transparent software artefact. A transparent software artefact would reduce the number of stakeholders' questions and lead to a more effective assessment of the software system. It would also reduce the number of communication channels needed for the sender to convey information to different stakeholders. This in turn would improve the software development process. Therefore, transparency is important to software engineering for helping stakeholders to see information about a software system. In the following sections, we summarise the contributions made and present some areas for future research.

8.2 Contributions

This research results in three major contributions. The first contribution is to explicate the concept of transparency in software engineering. We explore implications of transparency from different areas such as philosophy and business ethics as well as software engineering-related areas such as information privacy and agile development. Then we propose a working definition of transparency in software engineering and discuss our three attributes of transparency: accessibility, understandability, and relevance. The working definition and the three attributes of transparency are important to help us observe and interpret transparency's usefulness in software engineering. The working definition, once formalised, will help researchers to consolidate ideas relating to transparency in software engineering. It will also help software practitioners to articulate transparency problems during the software life cycle.

The second contribution of this thesis is to establish the importance of the concept of transparency in software engineering. We collect evidence about the concept of transparency and its relation with communication in software projects through the exploratory survey. This reveals that many communication problems are transparency problems in software projects. Moreover, we also demonstrate the importance of transparency in requirements engineering through the controlled experiment: a more transparent software artefact is more effective to receivers of information to answer questions than a less transparent software artefact. This helps us to gain confidence about the value of transparency in software engineering. This will also encourage researchers and software practitioners to think explicitly about transparency in software engineering.

The third contribution of this thesis is to start the validation process of our definition of transparency in software engineering. We start that process through the survey and the experiment. The survey enables us to evaluate our preliminary definition of transparency and the three attributes of transparency. The experiment enables us to apply our definition to analyse the degree of transparency of the requirements documents. The validation of our definition of transparency will help future researchers to improve what the concept of transparency should be in software engineering. It will also help future researchers to improve the structure of a diagnostic framework for software practitioners to articulate problems in communication during the software life cycle.

8.3 Future Work

We identify several potential areas for investigation from the discussion of our research findings. Much of the future work concerns our communication model, our definition, and our three attributes.

In the survey, responses to communication problems are different depending on the roles in our communication model. It will be interesting to investigate the asymmetry in the communication problems reported by our participants.

The survey shows that our participants are familiar with the term "transparency" in different contexts. However, it is unclear if our participants are also familiar with the application of existing notions of transparency in practice. An exploration of the application of existing notions of transparency in software engineering can be carried out. Such exploration will be useful to identify approaches that apply "transparency" in current practice. Future researchers may apply our definition of transparency to improve existing approaches. Important questions for the exploration are: Do stakeholders know how existing notions of transparency can be applied? Are they already applying transparency in practice?

In both the survey and the experiment, attributes such as accuracy and up-to-date information are mentioned by our participants as important. It will be interesting to investigate how other non-transparency attributes affect communication and/or transparency in software engineering.

In the experiment, our participants comment on different factors that affect the way they use the requirements documents. For future work on transparency, researchers can investigate how significant different factors affect accessibility, understandability, and relevance of information. For example, questions such as, "how do diagrams or pictures affect transparency?" can be investigated.

In addition, we have demonstrated the usefulness of transparency in requirements engineering but not in other aspects of software engineering such as project management, software design, software testing, and software maintenance. In future work on transparency, an investigation into how transparency benefits other aspects of software engineering can be conducted.

In this thesis, we assume in our working definition of transparency that receivers have reasonable expectations for information about a software system. The expectations depend on receivers' prior knowledge, which in turn affects how receivers perceive information presented in a communication channel. Therefore, how prior knowledge affects receivers' expectations of information can be investigated. Moreover, we also assume that receivers are reasonable stakeholders who have legitimate questions about a software system. In the real world, receivers can misuse the software system by asking questions that violate the security of the software system. It will be important to investigate the type of questions that different receivers have and what type of questions senders should or should not answer. Furthermore, we assume that the sender of information has no malicious intentions. However, this is not always true. The sender can give receivers distorted or false information about a software system. It will be important to investigate how distorted or false information affect the degree of transparency in software engineering.

Further to our attributes of transparency, accessibility, understandability, and relevance is future research investigating how each attribute of transparency relates to each other and how significant each attribute is to transparency. Such an investigation will improve the concept of transparency in software engineering.

Lastly, to apply the concept of transparency in practice, an investigation into how transparency can be measured in software engineering will be needed. In this thesis, we discuss the three attributes of transparency, which could be used as a part of the diagnostic framework. To measure each attribute, a GQM (goal, question, and metrics) can be used. Table 6.1 in Chapter 6 shows a preliminary structure of what the GQM might look like. The table suggests that each attribute of transparency is a goal and the questions are used to measure each attribute. The GQM as the diagnostic framework will be useful to software developers to articulate communication problems during the software life cycle. To improve the diagnostic framework, questions will need to be refined depending on the purpose of communication during the software life cycle. Furthermore, appropriate metrics will be required for answering each question in the diagnostic framework.

8.4 Final Thoughts

In this thesis, we explicate the concept of transparency and demonstrate its usefulness in one aspect of software engineering. The concept of transparency will be beneficial to software engineering as it will improve communication among stakeholders during the software life cycle. If software developers explicitly think about transparency, it is more likely that they communicate with other stakeholders successfully during software development. Moreover, as discussed in Chapter 1, too little transparency hinders communication during software development. However, we have not yet explored the case when there is too much transparency during software development. Too much transparency may incur a high cost and may hinder the software development process. This is because the sender of information may need to provide accessible, understandable, and relevant communication channels that accommodate different types of receivers. This can be costly as expert and non-expert receivers may require different types of information as well as different levels of detail about a software system. The next set of questions for research into transparency should be: How do we measure and control the degree of transparency in software engineering? How much transparency is enough for software development?

A
Draft Survey Design

Survey design

Goal definition for the survey -
Object of the study:
The objects are the RE process and system artefacts in software projects.

Purpose:
The purpose is to find any problems in software development concerning the RE process and to find aspects of transparency that are important to software development.

a) How do stakeholders get to know about the requirements of the software project? (Q4 - Q7)
   i. What type of information do they look for in the software project? e.g. features of the product or rationales for the decisions made. (Q4)
   ii. What procedures or system artefacts do the stakeholders use to obtain the information that they need? (Q5)
   iii. What are the problems in obtaining the information that the stakeholders need? (Q7)
b) How do stakeholders communicate with other stakeholders about the software project? (Q8 - Q11) Who do they generally communicate with? (Q8)
   i. What type of information do they convey? (Q~~8~~) Q9
   ii. What procedures or system artefacts do they use to communicate with other stakeholders? (Q~~9~~) Q10
   iii. What are the problems in communicating with other stakeholders? (~~Q11~~) Q12
c) Are there any techniques used to address the problems identified from the above? (Q~~12~~ – Q~~13~~)
   i. What are the techniques used? (Q~~12~~) Q13
   ii. How effective are the techniques in addressing these problems? (Q~~13~~) Q14

d) What is the general understanding of "transparency"? (Q~~14~~) Q15
e) Has the concept of transparency been considered in software development? (Q~~15~~) Q16
   i. Are there any terms that are specifically used to describe the concept of transparency in software development? (Q~~17~~) Q18
   ii. Are the problems identified from the above related to the concept of transparency? (Q~~18~~) Q20, Q21
   iii. Are there other problems in software development that are related to the concept of transparency? (Q~~19~~) Q20, Q21, Q22
   iv. What are the techniques used to address these problems? (~~Q20 – 21~~) Q26
f) Why should transparency be considered in software development?
   i. What are the attributes of transparency? (~~Q22 – Q23~~) Q23, Q24
   ii. What does transparency help in software development? (Q~~16~~) Q17
   iii. What should be transparent in the software project (what do stakeholders care about in the project)? (~~Q4, Q8~~) Q25, Q4, Q9

Quality focus:
The focus is to understand what transparency means to different stakeholders of software projects.

Perspective: the perspective is from different stakeholders (including software developers) of software projects.

Context: the study is run using stakeholders from different software projects.

B

Ethics Application for the Exploratory Survey

1. Completed Research Project Application Form to the University of Auckland Human Participants Ethics Committee.
2. Participant Information Sheet.
3. Email Invitation Template.

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Reference Number 2010 / __484___

University of Auckland Human Participants Ethics Committee (UAHPEC)

RESEARCH PROJECT APPLICATION FORM (2010)

DECLARATION FOR ALL SIGNATORIES:
The information supplied is, to the best of my knowledge and belief, accurate. I have read the current Guiding Principles and Applicants' Manual 2010. I clearly understand the obligations and the rights of the participants, particularly in regard to obtaining freely given informed consent. I have completed and submitted with this application the Application Checklist.

SUPERVISOR:

Name	Professor Clark Thomborson		
Postal address	Department of Computer Science, The University of Auckland, Private Bag 92019 Auckland New Zealand		
Email address	cthombor@cs.auckland.ac.nz		
Phone number	+64 9 373 7599 ext 85753		
Department	Department of Computer Science		
Signature		Date	

STUDENT (This includes Doctoral, Masters and Honours student): (If applicable)

Name	Yu-Cheng Tu		
Postal address	Department of Computer Science, The University of Auckland, Private Bag 92019 Auckland New Zealand		
Email address	ytu001@aucklanduni.ac.nz		
Phone number	+64 21 0471916		
Department	Department of Electrical and Computing Engineering		
Name of degree	Doctor of Philosophy in Electrical and Electronic Engineering		
Signature		Date	

OTHER INVESTIGATORS: (If applicable)

Names	Associate Professor Ewan Tempero		
Organisation	Department of Computer Science, The University of Auckland		
Is ethical approval being applied for from another institution?			NO

AUTHORISING SIGNATURES

Name of Head of Department or Nominee			
Email address			
Signature of Head of Department or Nominee		Date	

Name of Pro Vice Chancellor (Māori) / Nominee			
Email address			
Signature of Pro Vice Chancellor (Māori) / Nominee (If applicable)		Date	

APPLICATION CHECKLIST (Please delete whichever is not applicable)

General Information		
Is the application form dated for the current year? (Any application not using the current form on the website will be returned.)	YES	
Have you obtained all the signatures on pages 2 and 3 (wherever applicable)?	YES	
Have you addressed the ethical issues on A4? (Please note that "Not applicable" is not acceptable. The Committee will not consider the application if this is not answered adequately.)	YES	
Have you completed all sections?	YES	
Have you attached the advertisement?	YES	
Have you attached the questionnaire?	YES	
Have you attached the list of interview questions?	N/A	
Have you attached the transcriber confidentiality agreement? (Please refer to the Applicants' Manual Section 5c for sample format.)	N/A	
Have you consulted an ethics advisor in preparing the application? If yes, please provide the name and email address. Name of ethics advisor: ____ Email: ____		NO

Risk Assessment				
A.	**Risk of Harm**			
	1.	Does the research involve situations in which the researcher may be at risk of harm?		NO
	2.	Does the research involve the use of any method, whether anonymous or not, which might reasonably be expected to cause discomfort, pain, embarrassment, psychological or spiritual harm to the participants?		NO
	3.	Does the research involve processes that are potentially disadvantageous to a person or group, such as the collection of information which may expose the person/group to discrimination?		NO
	4.	Does the research involve collection of information about illegal behaviour(s) which could place the researcher or participants at risk of criminal or civil liability or be damaging to their financial standing, employability, professional or personal relationships?		NO
*	5.	Does the research involve any form of physically invasive procedure on participants, such as the collection of blood, body fluids, tissue samples, DNA, human tissue from a tissue bank, exercise or dietary regimes or physical examination?		NO
*	6.	Does the research involve any intervention administered to the participant, such as drugs, medicine (other than in the course of standard medical procedure), placebo, environmental conditions, food/drink?		NO
*	7	Does the research involve processes that involve EEG, ECG, MRI, TMS, FMRI, EMG, radiation, invasive or surface recordings?		NO
*	8.	Is the research considered a clinical trial?		NO
	9.	Does the research involve physical pain beyond mild discomfort?		NO
B.	**Informed and Voluntary Consent**			
	1.	Does the research involve participants giving oral consent rather than written consent? (If participants are anonymous the response is "No").		NO
	2.	Does the research involve participation of children (seven years old or younger)?		NO
	3.	Does the research involve participation of children under sixteen years of age where parental		NO

		consent is not being sought?		
	4.	Does the research involve participants who are in a dependent situation, such as people with a disability, residents of a hospital, nursing home or prison, or patients highly dependent on medical care?		NO
*	5.	Does the research involve participants who are being asked to comment on employers?		NO
	6.	Does the research involve participants whose capacity to give informed consent is in doubt?		NO
	7.	Does the research use previously collected information or biological samples for which there was no explicit consent?		NO
	8.	Is any part of the research conducted in University of Auckland class time?		NO
	9.	Does the research involve Focus Groups?		NO
C.	**Research conducted overseas**			
	1.	Will the research be conducted overseas?		NO
D.	**Privacy and confidentiality issues**			
	1.	Does the research involve evaluation of University of Auckland services or organisational practices where information of a personal nature may be collected and where participants may be identified?		NO
	2.	Does the research involve University of Auckland staff or students where information of a personal nature may be collected and where participants may be identified?		NO
	3.	Does the research involve matters of commercial sensitivity?		NO
E.	**Deception**			
	1.	Does the research involve deception of the participants, including concealment or covert observations?		NO
F.	**Conflict of interest**			
*	1.	Does the research involve a conflict of interest or the appearance of a conflict of interest for the researcher (for example, where the researcher is also the lecturer/teacher/treatment provider/colleague or employer of the participants, or where there is a power relationship between researcher and participants)?		NO
G.	**Cultural sensitivity**			
	1.	Does the research have impact on Maori persons as Maori?		NO
	2.	Does the research raise any specific ethnicity or cultural issues on any cultural groups other than Maori?		NO
H.	**Compensation to participants**			
	1.	Does the research involve payment or other financial inducements other than reasonable reimbursement of travel expenses or for time to participants?		NO
I.	**Procedural**			
	1.	Does the research involve a requirement imposed by an outside organisation for University of Auckland Human Participants Ethics Committee approval, for example a funding organisation or a journal, in which the researcher wishes to publish?		NO
	If none of the answers to the above Risk Assessment is "Yes" there is likelihood that the application will be considered as Low Risk Application. In this case, it will be reviewed immediately and you will hear the outcome within two weeks. If the application is not deemed low risk, it will be automatically put into the next agenda for Full Review. **NOTE: The Committee reserves the right to decide whether an application is low risk.**			

Have you included the following information in the Participant Information Sheet? **(Please note that this is not an exhaustive list. Please refer to the Applicants' Manual Sections 2c and 5a for more information.)**		
On University of Auckland Departmental letterhead	YES	
Project title	YES	
Researcher name	YES	
Position of staff and /or Degree of the student	YES	
Address by category, e.g. Participant Information Sheet for Manager	N/A	
Explain the project in simple language	YES	
Actual date/period for withdrawal of data	YES	
Length of time involvement	YES	
Source of funding	N/A	
State whether audio/videotaping	N/A	
Data storage/retention/destruction/future use	YES	
Confidentiality statement	YES	
Participation/non-participation statement (Please refer to the Applicants' Manual Section 2c iv)	YES	
Contact details (This includes the details of the researcher, supervisor, HOD and Chair)	YES	
Approval wording	YES	

Have you included the following information in the Consent Form? **(Please note that this is not an exhaustive list. Please refer to the Applicants' Manual Sections 2d and 5b for more information.)**		
On University of Auckland Departmental letterhead	YES	
Project title	YES	
Researcher name	YES	
Address by category, e.g. Consent Form from Manager	N/A	
Actual date/period for withdrawal of data	YES	
Length of time involvement	YES	
Consent for audio/videotaping	N/A	
Data storage/retention/destruction/future use	YES	
Confidentiality statement	YES	
Participation/non-participation statement	YES	
Participant's and / or legal guardian's name, signature and date	N/A	
Approval wording	YES	

SECTION A:

1. Project title

Transparency in software development

2. Aims/objectives of project
(Describe in plain language that is comprehensible to lay people and free from jargon.)

The main objective of this research project is to formulate a theory of transparency in software engineering. We would like to know if the concept of transparency has been considered by people involved in software projects and to find any software engineering problems that are related to transparency. We also aim to understand what transparency means in software engineering and to identify aspects of transparency that are important in practice.

To achieve our objectives, we aim to answer the following research questions:
- **Has the concept of transparency been considered in the software industry?**
- **What are the problems in software engineering that are related to transparency?**
- **What is transparency in software engineering?**
- **What are the properties of transparency in software engineering?**
- **What should be transparent in software engineering?**

3. Research background
(Provide sufficient information to place the project in perspective and to allow the significance of the project to be assessed.)

The term "transparency" has appeared in different research areas such as government, business, and ethics. Transparency in many research areas refers to the quality of a process, or information being easily understood or recognised. It is also about making information accessible to stakeholders and enabling better decision-making [1]. Moreover, transparency is important for protecting stakeholder interests as well as enabling communication with stakeholders [2]. These notions of transparency are important to software engineering, because stakeholder communication is one of the key factors to the success of software projects. Poor communication between software developers and stakeholders would hinder the process for identifying what stakeholders want for the software system [3]. Poor communication would also result in misunderstanding of requirements of the software system and inaccurate documentation [4].

One way to prevent failures in stakeholder communication is to disclose all information relevant to the software system. However, this approach is limited. The information disclosed usually contains data of every variety, ranging from general system overview to detailed technical matters. The information disclosed might become unintelligible to stakeholders who are not experts in the area. The information might also become hard to comprehend when there is too much information available. These problems often appear in large infrastructure projects that involve public participation. To overcome these problems, information about the software system should be made accessible, relevant, and understandable to various stakeholders. The concept of transparency is therefore important for resolving these problems.

However, based on our preliminary literature review, we found only a few articles that discussed the ideas of transparency in software engineering or related field. It seems that transparency is not a well-known concept in software engineering. Therefore in this research project, we plan to study the views of software industry on the concept of transparency. We also aim to identify any problems and solutions in practice relating to transparency.

References
[1] R. W. Oliver, What is transparency? McGraw-Hill, 2004.

[2] A. Vaccaro and P. Madsen, "Corporate dynamic transparency: the new ict-driven ethics?" Ethics and Inf. Technol., vol. 11, no. 2, pp. 113-122, 2009.

[3] H. Saiedian and R. Dale, "Requirements engineering: making the connection between the software developer and customer," Information and Software Technology, vol. 42, pp. 419-428, 2000.

[4] J. Coughlan and R. D. Macredie, "Effective communication in requirements elicitation: a comparison of methodologies," Requirements Engineering, vol. 7, no.2, pp. 47-60, 2002.

4. **Identify the ethical issues arising from this project and explain how they can be resolved.**
(For example: confidentiality, anonymity, informed consent, participant's rights to withdraw, conflict of interest, etc.)
(UAHPEC expects applicants to identify the ethical issues in the project and explain in the documentation how they have been resolved. The application will not be considered if this is not answered adequately. A "Not applicable" response is not acceptable.)

Confidentiality: We are unable to make guarantees to confidentiality of the data collected from the participants. This is because we will be using a web-based questionnaire for the research project, in which we have little control of data privacy on the Internet. There is some small risk of exposing data collected from the participants to other parties on the Internet. To mitigate this risk, all data collected will not be available on the Internet. Access to the data will be limited to the supervisors and the student of this research project.

Anonymity: To help protect participant's privacy on the Internet, we will not collect any information that can be used to identify the participants. Moreover, we will not collect information that will identify other individuals or organisations that the participants work for. This also helps us to protect anonymity of the data collected from the participants. However, there are possibilities that the participants might accidentally reveal any personal or organisational information. To reduce the likelihood, participants will be made aware of the risks before consenting to participate in the research. We will also ask the participants not to provide any identifying information in the responses. We will remove any identifying information disclosed from the responses. In addition, the information provided by the participants will be analysed and reported anonymously.

Since we will be using a web-based questionnaire for the research, it is possible for us to identify participants by their IP addresses. To help protect anonymity of participation, we will not track IP addresses of the participants. Moreover, we will not ask the participants for their email addresses or any information that directly reveal their identities.

Rights to withdraw: Participant's rights to withdraw data from the research would not be possible, because the data is anonymous. We will not be able to identify individual participants and their responses. Participants will be made aware that they are unable to withdraw data after submitting the questionnaire in the PIS. However, participants are entitled to withdraw from involvement in the research at any time before submitting the questionnaire.

Informed consent: Participants will not be required to sign consent forms due to anonymous responses. However, we will include a consent page at the beginning of the questionnaire that enables participants to indicate if they understand what is involved in the research. We will also note to the participants that by submitting the questionnaire indicates that they agree to participate in the research.

SECTION B:

1. **Who are the participants in the research?**
(Delete those who do not apply)

Adults	

2. **Explain how many organisations, departments within the organisations, and individuals you wish to recruit.**
(Attach any letter of support you may have had from an organisation.)

We wish to recruit as many individuals (who are or have been involved in software project) as we can.

3. **How will you obtain the names and contacts of participants?**
(If by advertisement or email, attach a copy to the application. If through an agency holding these details, attach a copy of support letter.)

Recruitment of participants will occur via email. Email invitations will be sent to potential participants by using the student's (Yu-Cheng Tu) personal contacts. A mailing list of professional software engineers will also be used with the permission of the mailing list owner and within the privacy declaration on the mailing list. Names and contacts of participants will not be obtained.

4. Who will make the initial approach to potential participants?
(For example: will the owner of the database send out letters?)

The student (Yu-Cheng Tu) will make the initial approach by sending invitation for research participation to her personal contacts via email. The student will also ask the owner of the mailing list to forward the email for invitation to research participation.

5. Is there any special relationship between participants and researchers?

	NO

6. Are there any potential participants who will be excluded?

	NO

SECTION C: RESEARCH PROCEDURES

1. **Project duration** (Dates during which data needs to be collected for this study and requires ethics approval.)

From ______01/10/2010________ to _______30/09/2012______

2. Describe the study design.
(For example: If it is a longitudinal study, explain what a longitudinal study is and provide the details.)

This study is a retrospective study, which involves people from different software projects to look back at their experience in software engineering. The study will consist of two main parts. The first part of the study will ask participants to identify any problems related to information gathering and communication that they have encountered in software projects. The second part of the study will be asking participants to relate their knowledge and experience in software engineering with the concept of transparency. The study will also ask participants a few general questions about their roles in software projects as well as their level of knowledge and experience in software engineering.

3. List all the methods used for obtaining information.
(Delete those that do not apply)

Questionnaires (attach questionnaire)	

4. Who will carry out the research procedures?

The student (Yu-Cheng Tu).

5. a) Where will the research procedures take place?

(Physical location/setting)

The research procedures will take place using web-based questionnaire.

b) If the study is based overseas, which countries are involved?
(Provide local contact information on the PIS.)

N/A

c) If the study is based overseas, explain what special circumstances arise and how they will be dealt with? Explain if there are any special requirements of the country (e.g. research visa) and/or the community with which the research will be carried out?

N/A

6. If the questionnaire is web-based, explain how anonymity can be preserved.
(Indicate this on the PIS.)

To help protect the anonymity of participation, IP addresses will not be tracked or saved in the questionnaire. Moreover, we will not ask for any information such as email addresses that can lead to the identification of the participants in the questionnaire. We will remove any personal or organisational information revealed from participants' responses. In addition, the results will be assessed and reported anonymously.

7. How much time will participants need to give to the research?
(Indicate this on the PIS.)

Approximately 30 minutes.

8. Will information on the participants be obtained from third parties?

	NO

9. Will any identifiable information on the participants be given to third parties?

	NO

10. Are you intending to conduct the research in University of Auckland class time?

	NO

11. Is deception involved at any stage of the research?

	NO

12. Is there any koha, compensation or reimbursement of expenses to be made to participants?

	NO

13. a) Does the research involve the administration of any substance to participants?

	NO

b) Does this research involve potentially hazardous substances?

	NO

SECTION D: INFORMATION AND CONSENT

1. By whom and how will information about the research be given to participants?
(For example: in writing or verbally – a copy of information to be given to prospective participants in the form of a PIS must be attached to this application.)

The student (Yu-Cheng Tu) will attach the PIS when sending invitation emails to prospective participants. There will be a consent page at the beginning of the questionnaire.

2. a) Will the participants have difficulty giving informed consent on their own behalf?
(Consider physical or mental condition, age, language, legal status, or other barriers.)

	NO

b) If participants are not competent to give fully informed consent, who will consent on their behalf?
(For example: parents/guardians)

N/A

3. a) If a questionnaire is used, will the participants have difficulty completing the questionnaire on their own behalf?
(Consider physical or mental condition, age, language, legal status, or other barriers.)

	NO

b) If participants are not competent to complete the questionnaire, who will act on their behalf?
(For example: parents/guardians)

N/A

4. Is informed consent obtained in writing?

	NO (Explain, justify and indicate in the PIS.) Because we will be collecting anonymous responses which require completed questionnaire as evidence of informed consent.

5. Is access to the Consent Forms restricted to the Principal Investigator and/or the researcher?

YES	

6. **Will Consent Forms be stored by the Principal Investigator, in a locked cabinet, on University premises?**

YES	

7. **Are Consent Forms stored separately from data and kept for six years?**

	NO (Explain, justify and indicate in the PIS.) Because participants will not be required to sign consent forms separately. Completed questionnaire will be treated as consent forms.

SECTION E: STORAGE AND USE OF RESULTS

1. **Will the participants be audio-taped, video-taped, or recorded by any other electronic means such as Digital Voice Recorders?**
(Explain in the PIS and CF. Consider whether recording is an optional or necessary part of the research design, and reflect this in the CF.)

	NO

2. **a) Will the recording be transcribed or translated?**

	NO

b) Who will be transcribing the recordings?
(If someone other than the researcher is the transcriber, attach a copy of the Confidentiality Agreement and indicate in the PIS and CF.)

N/A

c) If recordings are made, will participants be offered the opportunity to edit the transcripts of the recordings?

N/A

d) Will participants be offered their tapes or files of their recording (or a copy thereof)?

N/A

3. **If a questionnaire is used, please explain if there is any coding.**

	NO

4. **a) Explain how and how long the data (including audio-tapes, video-tapes, digital voice recorder, and electronic data) will be stored.**
(Indicate this in the PIS. The period data is to be kept will be commensurate to the scale of its research. For peer reviewed publication that might be further developed, the University expects six years.)

All data provided by the participants will not be made available on the Internet. Data will be used and stored until the completion of the PhD research of the student.

b) Explain how data will be used.
(Indicate this in the PIS.)

Data will be analysed and reported anonymously for the PhD research of the student (Yu-Cheng Tu). Findings of this research will be reported in the student's PhD thesis. Data may also be used in conference papers for reporting the findings of this research project.

c) Explain how data will be destroyed.
(Indicate this in the PIS.)

All data provided by the participants will be deleted after the completion of the PhD research of the student.

5. **Describe any arrangements to make results available to participants.**
(Explain this in the PIS.)

The raw data will not be made available to participants and the participants will not be able to withdraw any information that they had provided after submitting the questionnaire. However, participants are able to withdraw from the involvement in the research at any time before submitting the questionnaire. A summary of the research findings will be made available online.

6. **a) Are you going to use the names of the research participants in any publication or report about the research?**
(The PIS must inform the participants, and be part of the consent obtained in the CF.)

	NO

b) If you don't use their names, is there any possibility that individuals or groups could be identified in the final publication or report?
(This is a problem either when one is dealing with a small group of participants known to a wider public or when there is to be a report back to participants likely to know each other.)

	NO

SECTION F: TREATY OF WAITANGI

1. **Does the proposed research have impact on Māori persons as Māori?**

	NO (Go to Section G.)

2. **Explain how the intended research process is consistent with the provisions of the Treaty of Waitangi.**
(Refer to the Applicants' Manual 2010 for further information.)

3. **Identify the group(s) with whom consultation has taken place, describe the consultation process, and attach evidence of the support of the group(s).**

4. Describe any on-going involvement the group(s) consulted has/have in the project.

5. Describe how information will be disseminated to participants and the group consulted at the end of the project.

SECTION G: OTHER CULTURAL ISSUES

1. Are there any aspects of the research that might raise any specific cultural issues?

	NO (Go to Section H)

2. What ethnic or cultural group(s) does/do the research involve?

3. Identify the group(s) with whom consultation has taken place, describe the consultation process, and attach evidence of the support of the group(s).

4. Describe any on-going involvement the group(s) consulted has/have in the project.

5. Describe how information will be disseminated to participants and the group(s) consulted at the end of the project.

SECTION H: CLINICAL TRIALS

1. Is this project a Clinical Trial?

	NO (Go to Section I)

2. Is this project initiated by a Pharmaceutical Company?

YES	NO

3. **Are there other NZ or International Centres involved?**

YES	NO

4. **Is there a clear statement about indemnity?**

YES	NO

5. **Is Standing Committee on Therapeutic Trials (SCOTT) approval required?**

YES (Attach)	NO

6. **Is National Radiation Laboratory approval required?**

YES (Attach)	NO

7. **Is Gene Therapy Advisory Committee on Assisted Human Reproduction (NACHDSE) approval required?**

YES (Attach)	NO

SECTION I: RISKS AND BENEFITS

1. **What are the possible benefits to research participants of taking part in the research?**

There is no direct benefit to the participants taking part in this research. However, a summary of the research findings will be made available online.

There will be benefits to the software engineering community. The results of this research will help researchers to understand how the software industry views the concept of transparency. The results will also help the researchers to formulate a definition of transparency for software engineering. Moreover, the results will benefit the software industry by making transparency problems explicit in software practice. This will help the software industry to improve the quality of the software engineering process as well as the information provided to people involved in software projects.

2. **What are the possible risks to research participants of taking part in the research?**
(Make sure that you have clearly identified/explained these risks in the PIS and CF(s).)

Because of the use of web-based questionnaire, there is a small likelihood that the data provided by participants might be exposed to other parties on the Internet. To minimise the likelihood, we will not make the data publicly available on the Internet. Access to the data will be limited to the supervisors and student of this research project.

Since there is a small likelihood that the data might be exposed to other parties on the Internet, there will be some small risk to the privacy of the participants. Participants might reveal their identities or the organisations that they work for in the responses. This might also put the participants at risks of losing their jobs if the responses contain information that might damage the reputation of organisations. To mitigate the risks, we will not ask for any information that can be used to identify the participants. We will not collect any information that will identify other individuals or organisations that the participants work for. Moreover, we will not track the IP addresses of the participants. However, it is possible that the participants might accidentally reveal any personal or organisational information. To reduce the likelihood, participants will be made aware of the risks before consenting to participate in the research. We will also ask the participants not to give any identifying information in their responses. We will remove any personal or organisational information disclosed. In addition, the results will be analysed and reported anonymously.

3. a) Are the participants likely to experience discomfort (physical, psychological, social) or incapacity as a result of the procedures?

	NO

b) What other risks are there?

N/A

c) What qualified personnel will be available to deal with adverse consequences or physical or psychological risks?
(Explain in the PIS.)

N/A

SECTION J: FUNDING

1. Have you applied for, or received funding for this project?

	NO (Proceed to Section K)

2. From which funding bodies? (Quote the contract reference number.)

3. Is this a UniServices project?

YES (Quote the contract reference number.)	NO

4. Explain investigator's financial interest, if any, in the outcome of the project.

5. Do you see any conflict of interest between the interests of the researcher, the participants or the funding body?

YES (Explain.)	NO

SECTION K: HUMAN REMAINS, TISSUE AND BODY FLUIDS

1. Are human remains, tissue, or body fluids being used in this research?

	NO (Go to Section L.)

2. How will the material be taken?
(For example: at operation, urine samples, archaeological digs, autopsy.)

3. Is the material being taken at autopsy?

YES	NO

4. Is material derived or recovered from archaeological excavation?

YES (Explain how the wishes of Iwi and Hapū (descent groups), or similar interested persons, or groups, have been respected?)	NO

5. Will specimens be retained for possible future use?

YES (Explain and state this in the PIS.)	NO

a) Where will the material be stored?

b) How long will it be stored for?

6. a) Will material remain after the research process?

YES (Explain and state this in the PIS.)	NO

c) How will material be disposed of?
(Explain how the wishes with regard to the disposal of human remains of the whanau (extended family) of similar interested persons will be respected.)

d) Will material be disposed of in consultation with relevant cultural groups?

YES (Explain and state this in the PIS.)	NO

7. Is blood being collected?

YES (Complete this section and state in the PIS.)	NO

a) What is the volume at each collection?

b) How frequent are the collections?

c) Who is collecting it?

d) Explain how long it will be kept and how it will be stored.

e) Explain how it will be disposed of.

SECTION L: OTHER INFORMATION

1. **Have you made any other related applications?**

	NO

2. **If there is relevant information from past applications or interaction with UAHPEC, please indicate and attach.**

N/A

3. **Are there any other matters you would like to raise that will help the Committee review your application?**

	NO

--- END OF APPLICATION FORM ---

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Department of Electrical and Computing Engineering
The University of Auckland
Private Bag 92019
Auckland, New Zealand
Phone: +64 9 373 7599 ext 88158

Participant Information Sheet

Project title: Transparency in software development

Researchers: Yu-Cheng Tu / Professor Clark Thomborson / Associate Professor Ewan Tempero

To: potential participant

This research is being undertaken as a part of a PhD degree at the Department of Electrical and Computing Engineering, University of Auckland by Yu-Cheng Tu. The purpose of this research is to study the views of people involved in software projects on the concept of transparency. This research also aims to study how people gather information and communicate in software projects, and to identify any relevant issues that arise from information gathering and communication in software projects.

Anyone who is directly or indirectly involved in software projects is invited to participate in this research. Participation in this research project is voluntary, and you may choose not to participate.

Our research involves the use of an online questionnaire, which will take approximately 30 minutes to complete. The questionnaire will consist of two main parts. In the first part of the questionnaire, you will be asked about how you gather information and communicate with other people in software projects. You will also be asked about the problems that you or other people have encountered in software projects. In the second part, you will be asked some questions about the concept of transparency and how important it is to software engineering.

The data that you provide will be used for the PhD research of the student (Yu-Cheng Tu). Data may also be used to report findings of this research in conference papers. Findings of this research will be reported in the student's PhD thesis. All data collected in this research will remain anonymous. Any identifying information such as your name, email and IP address will not be collected. Data will not be made publicly available on the Internet. They will only be available to the researchers of this research project, and will be stored until the completion of the PhD research of the student.

If you decide to participate in this research, you have the right to withdraw from participation at any time before the point of submitting the questionnaire. However, due to the nature of anonymous responses, you are unable to withdraw data provided by you from the research after the point of submitting the questionnaire.

We will do our best to keep your information confidential. Please note that there is always some small risk of exposing data on the Internet, in which your privacy might be breached. To help protect your privacy, you will not be asked to provide any information that will personally identify you. In addition, we also ask you not to provide any information in your responses that could lead to the identification of individuals or organisations. This is to help protect you and your organisation from harm. However, if you accidentally or mistakenly revealed any personal or organisational information in your responses, this information will be removed. The information that you provide will be analysed and reported anonymously.

If you are willing to participate, please complete the online questionnaire at http://www.cs.auckland.ac.nz/research/groups/ssg/homepages/yu-cheng/questionnaire.html. You will not be asked to sign a consent form. However, there will be an electronic consent at the beginning of the online questionnaire. Please note that by submitting the online questionnaire indicates that you agree to take part in this research.

A summary of the research findings will be made available online at http://www.cs.auckland.ac.nz/research/groups/ssg/homepages/yu-cheng/summary.html after the completion of this research project. If you have any questions about the research, please contact us. Contact details are provided below.

Contacts

Professor Allan Williamson (Head of Department, Department of Electrical and Computing Engineering)
Phone: +64 9 373 7599 ext 87922
Email: ag.williamson@auckland.ac.nz

Professor Clark Thomborson (Supervisor, Department of Computer Science)
Phone: +64 9 373 7599 ext 85753
Email: cthombor@cs.auckland.ac.nz

Associate Professor Ewan Tempero (Supervisor, Department of Computer Science)
Phone: +64 9 373 7599 ext 83765
Email: e.tempero@cs.auckland.ac.nz

Yu-Cheng Tu (PhD student)
Phone: +64 21 0471916
Email: ytu001@aucklanduni.ac.nz

For any queries regarding ethical concerns you may contact the Chair, The University of Auckland Human Participants Ethics committee, The University of Auckland, Office of the Vice Chancellor, Private Bag 92019, Auckland 1142. Telephone 09 373-7599 extn. 83711.

APPROVED BY THE UNIVERSITY OF AUCKLAND HUMAN PARTICIPANTS ETHICS COMMITTEE ON 30 September 2010 FOR (3) years, Reference Number 2010/484

Email invitation to participate in the research.

Subject: Invitation to participate in a PhD research project at the University of Auckland

Hi,

My name is Yu-Cheng Tu, and I am a PhD student at the University of Auckland. I would like to invite anyone who is directly or indirectly involved in software projects to take part in my research. The purpose of my research is to study what the concept of transparency means to the people involved in software projects. I am also aiming to study how people gather information and communicate in software projects.

The research involves the use of an online questionnaire, which will take approximately 30 minutes to complete. In the questionnaire, you will be asked some questions about your experience in software projects and your knowledge about the concept of transparency.

Your participation in this research is voluntary. You may choose not to participate. However, your participation and feedback will be of great value to my research. The results will help my research in understanding how the software industry views the concept of transparency as well as formulating a definition of transparency in software engineering. This in turn will help to improve the quality of the software engineering process.

Attached is a copy of the Participant Information Sheet, which describes this research in detail. Please read it carefully. If you are willing to participate, please complete the online questionnaire at http://www.cs.auckland.ac.nz/research/groups/ssg/homepages/yu-cheng/questionnaire.html.

A summary of the research findings will be made available online at http://www.cs.auckland.ac.nz/research/groups/ssg/homepages/yu-cheng/summary.html after the completion of this research project. If you have any questions about the research, you can contact me at ytu001@aucklanduni.ac.nz. Contact details are also provided in the Participant Information Sheet.

Thank you for considering this request.

Sincerely,
Yu-Cheng Tu

PhD Student
Department of Electrical and Computing Engineering
University of Auckland
Supervisors: Prof. Clark Thomborson, Assoc. Prof. Ewan Tempero (Department of Computer Science)

C
Web-based Questionnaire for the Exploratory Survey

Transparency in software development

1. Participants Consent Form

Project title: Transparency in software development

Researchers: Yu-Cheng Tu / Professor Clark Thomborson / Associate Professor Ewan Tempero

I have read the Participant Information Sheet, have understood the nature of the research and why I have been selected. I have had the opportunity to ask questions and have them answered to my satisfaction.

- I agree to take part in this research.

- I understand that participation in this research project is voluntary.

- I understand that I am free to withdraw participation at any time before the point of submitting the questionnaire.

- I understand that I will not be able to withdraw any data provided by me after the point of submitting the questionnaire.

- I understand that the online questionnaire will take approximately 30 minutes.

- I understand that my responses will remain anonymous and any identifying information such as name, email address, and IP address will not be collected.

- I understand the risks of losing data privacy when using an online questionnaire.

- I understand that data will be stored and used for the PhD research of the student (Yu-Cheng Tu) until the completion of the PhD research.

*** ELECTRONIC CONSENT: please select your choice below.**
Click on the "agree" button below indicates that:
- You are aged 16 years or older, and
- You have read the above information, and
- You understand that, by submitting this questionnaire electronically you agree to take part in this research.

If you do not wish to participate in the research project, please decline by clicking on the "disagree" button.

○ agree

○ disagree

APPROVED BY THE UNIVERSITY OF AUCKLAND HUMAN PARTICIPANTS ETHICS COMMITTEE ON 30 September 2010 FOR 3 YEARS, REFERENCE NUMBER 2010/484

Transparency in software development

There are 28 questions in this questionnaire, and it is not compulsory to fill in any questions you do not wish to. To progress through this questionnaire, please use the following navigation buttons:

- Click the "Next" button to continue to the next page.
- Click the "Prev" button to return to the previous page.
- Click "Exit this questionnaire" if you wish to exit.
- Click the "Submit" button (on the last page) to complete the questionnaire and exit.

Please note that when the "Next" button is clicked any responses made on the page will be saved.

Please click the "Submit" button (on the last page) to complete the questionnaire. Any responses made without clicking the "Submit" button will be treated as withdrawing from participating in the research.

For more information about the web-based survey host, please see SurveyMonkey at http://www.surveymonkey.com/, and it's Privacy Policy and Security Statement.

Transparency in software development

2. Demographics

1. What aspects of the software project are you involved in? (Select all that apply)

- [ ] Software requirements
- [ ] Software design
- [ ] Software development
- [ ] Software testing
- [ ] Software maintenance
- [ ] Software configuration management
- [ ] Project management
- [ ] Quality management
- [ ] Other (please specify)

2. What roles do you generally have in the software project? (Select all that apply)

- [ ] Requirements engineer
- [ ] Developer
- [ ] Architect
- [ ] Project manager
- [ ] User / user representative
- [ ] Client
- [ ] Regulator
- [ ] Other (please specify)

3. How well do you believe that your knowledge or experience is in...

	Very good	Good	Average	Poor	Very poor
The software project?	○	○	○	○	○
Requirements engineering?	○	○	○	○	○
Communicating with different stakeholders? (A stakeholder can be anyone who is involved in the software project, e.g. users, market analysts, regulators, software engineers)	○	○	○	○	○
Software engineering?	○	○	○	○	○

Transparency in software development

3. Gathering information and communication in software projects

4. What type of information do you look for in the software project? (Select the three most important)

- [ ] Business objectives
- [ ] User requirements
- [ ] System specification
- [ ] Software architecture
- [ ] Design rationale
- [ ] Bug report
- [ ] User manual
- [ ] Cost analysis
- [ ] Risk analysis
- [ ] Market research
- [ ] Other (please specify)

5. How do you get to know the information in the software project?
For each statement below, choose 'Always' if the statement is true more than 90% of the time, 'Frequently' if it is true 50%-90% of the time, 'Seldom' if it is true less than 50% of the time and 'Never' if the statement is not true.

	Always	Frequently	Seldom	Never
I consult comprehensive documentation.	◯	◯	◯	◯
I consult informal documentation.	◯	◯	◯	◯
I learn about the information at planning meetings.	◯	◯	◯	◯
I learn about the information by informal discussions with other members of my organisation.	◯	◯	◯	◯
I learn about the information by informal discussions with clients.	◯	◯	◯	◯
I innovate based on my knowledge of the problem domain.	◯	◯	◯	◯
I am given the information that I need.	◯	◯	◯	◯
I have to search for the information that I need.	◯	◯	◯	◯
I have to interpret the documentation I consult.	◯	◯	◯	◯

Other (please describe)

Transparency in software development

6. How well do you think the following ways are in helping you to know the information in the software project?

	Very good	Good	Satisfactory	Poor	Very poor	N/A
I consult comprehensive documentation.	◯	◯	◯	◯	◯	◯
I consult informal documentation.	◯	◯	◯	◯	◯	◯
I learn about the information at planning meetings.	◯	◯	◯	◯	◯	◯
I learn about the information by informal discussions with other members of my organisation.	◯	◯	◯	◯	◯	◯
I learn about the information by informal discussions with clients.	◯	◯	◯	◯	◯	◯
I innovate based on my knowledge of the problem domain.	◯	◯	◯	◯	◯	◯
I am given the information that I need.	◯	◯	◯	◯	◯	◯
I have to search for the information that I need.	◯	◯	◯	◯	◯	◯
I have to interpret the documentation I consult.	◯	◯	◯	◯	◯	◯

Other (please comment on how well you think other ways are in helping you)

7. What problems do you encounter when trying to know the information in the software project?
For each problem below, choose 'Always' if you encounter the problem more than 90% of the time, 'Frequently' if you encounter the problem 50%-90% of the time, 'Seldom' if you encounter the problem less than 50% of the time and 'Never' if you never encounter the problem.

	Always	Frequently	Seldom	Never
There are too many managers and clients to deal with.	◯	◯	◯	◯
The information is difficult to understand.	◯	◯	◯	◯
I don't know what information to look for in the software project.	◯	◯	◯	◯
I can't find the information or it is difficult to obtain the information that I need.	◯	◯	◯	◯
The information contains errors.	◯	◯	◯	◯
The given information is not what I need.	◯	◯	◯	◯

Other (please describe)

Transparency in software development

8. What type of stakeholder do you generally communicate with in the software project?
A stakeholder can be anyone who is involved in the software project, e.g. users, market analysts, software engineers.
(Select the three most important)

- ☐ Requirements engineer
- ☐ Developer
- ☐ Architect
- ☐ Project manager
- ☐ User / user representative
- ☐ Client
- ☐ Regulator
- ☐ Other (please specify)

9. What type of information about the software project do you convey to other stakeholders?
A stakeholder can be anyone who is involved in the software project, e.g. users, market analysts, software engineers.
(Select the three most used)

- ☐ Business objectives
- ☐ User requirements
- ☐ System specification
- ☐ Software architecture
- ☐ Design rationale
- ☐ Bug report
- ☐ User manual
- ☐ Cost analysis
- ☐ Risk analysis
- ☐ Market research
- ☐ Other (please specify)

Transparency in software development

10. How do you communicate with other stakeholders about the software project? For each statement below, choose 'Always' if the statement is true more than 90% of the time, 'Frequently' if it is true 50%-90% of the time, 'Seldom' if it is true less than 50% of the time and 'Never' if the statement is not true.

	Always	Frequently	Seldom	Never
I ask other stakeholders to consult comprehensive documentation.	○	○	○	○
I ask other stakeholders to consult informal documentation.	○	○	○	○
I give the information to other stakeholders at planning meetings.	○	○	○	○
I give information about the project by informal discussions with other members of my organisation.	○	○	○	○
I give information about the project by informal discussions with clients.	○	○	○	○
I don't communicate with other stakeholders.	○	○	○	○

Other (please descirbe)

11. How well do you think the following ways are in helping you to communicate with other stakeholders?

	Very good	Good	Satisfactory	Poor	Very poor	N/A
I ask other stakeholders to consult comprehensive documentation.	○	○	○	○	○	○
I ask other stakeholders to consult informal documentation.	○	○	○	○	○	○
I give the information to other stakeholders at planning meetings.	○	○	○	○	○	○
I give information about the project by informal discussions with other members of my organisation.	○	○	○	○	○	○
I give information about the project by informal discussions with clients.	○	○	○	○	○	○
I don't communicate with other stakeholders.	○	○	○	○	○	○

Other (please comment on how well you think other ways are in helping you)

Transparency in software development

12. What problems do you encounter when you communicate with other stakeholders?
For each problem below, choose 'Always' if you encounter the problem more than 90% of the time, 'Frequently' if you encounter the problem 50%-90% of the time, 'Seldom' if you encounter the problem less than 50% of the time and 'Never' if you never encounter the problem.

	Always	Frequently	Seldom	Never
There are too many managers and clients to deal with.	○	○	○	○
The information is difficult to understand for other stakeholders.	○	○	○	○
I don't know what information to give to other stakeholders.	○	○	○	○
Other stakeholders can't find the information or it is difficult to obtain the information.	○	○	○	○
The information contains errors.	○	○	○	○
The information given to the stakeholders is not what they need.	○	○	○	○

Other (please describe)

13. What techniques do you use to help you in overcoming the problems encountered during communication with other stakeholders?
For each statement below, choose 'Always' if you use the technique more than 90% of the time, 'Frequently' if you use the technique 50%-90% of the time, 'Seldom' if you use the technique less than 50% of the time and 'Never' if you never use the technique.

	Always	Frequently	Seldom	Never
Use of prototypes.	○	○	○	○
Use of diagrams, e.g. use case diagrams.	○	○	○	○
Use of formal methods, e.g. formal notations such as Z.	○	○	○	○
Have regular meetings/discussions with other stakeholders.	○	○	○	○
Have a communication channel that allows for continuous feedback, e.g. use of emails.	○	○	○	○

Other (please describe other techniques you used).

Transparency in software development

14. How effective are the techniques in helping you to overcome the problems encountered during communication with other stakeholders?

	Very good	Good	Satisfactory	Poor	Very poor	N/A
Use of prototypes.	◯	◯	◯	◯	◯	◯
Use of diagrams, e.g. use case diagrams.	◯	◯	◯	◯	◯	◯
Use of formal methods, e.g. formal notations such as Z.	◯	◯	◯	◯	◯	◯
Have regular meetings/discussions with other stakeholders.	◯	◯	◯	◯	◯	◯
Have a communication channel that allows for continuous feedback, e.g. use of emails.	◯	◯	◯	◯	◯	◯

Other (please comment on how effective other techniques are)

Transparency in software development

4. Transparency in software engineering

15. Are you familiar with the term "transparency" used in the context of...? (Select all that apply)

- [ ] I am not familiar with the term "transparency" used in any context.
- [ ] Philosophy (e.g. referential transparency, epistemic transparency).
- [ ] Government, business, and ethics (e.g. transparency implies openness, and accountability of organisations).
- [ ] Public participation (e.g. principle of making participation process and its outcome clear to the public).
- [ ] Computing (e.g. network property that makes users unaware that they are interacting with the network).
- [ ] Other (please describe)

We define transparency in software engineering as:

Enabling stakeholders to answer their questions about the software project.

A stakeholder can be anyone involved in the software project, e.g. users, market analysts, software engineers.

---

16. Has this concept of transparency been considered in software engineering?

- ( ) Yes
- ( ) No
- ( ) Don't know

17. Do you think this concept of transparency is important for...? (Select all that apply)

- [ ] Helping you to know about the software project.
- [ ] Helping you to communicate with other stakeholders.
- [ ] Other (please describe)

Transparency in software development

18. Are there any other terms used in software engineering to describe this concept of transparency?

- ○ Yes
- ○ No
- ○ Don't know

If yes, please specify the terms used.

19. To whom do you think this concept of transparency is required? (Select all that apply)

- ☐ Requirements engineer
- ☐ Developer
- ☐ Architect
- ☐ Project manager
- ☐ User / user representative
- ☐ Client
- ☐ Regulator
- ☐ Other (please specify)

20. Which of the following problems that you encountered when trying to know the information in the software project do you think are related to this concept of transparency?
(Select all that apply)

- ☐ There are too many managers and clients to deal with.
- ☐ The information is difficult to understand.
- ☐ I don't know what information to look for in the software project.
- ☐ I can't find the information or it is difficult to obtain the information that I need.
- ☐ The information contains errors.
- ☐ The given information is not what I need.
- ☐ Other (please describe)

Transparency in software development

21. Which of the following problems that you encountered when communicating with other stakeholders do you think are related to this concept of transparency? (Select all that apply)

- [ ] There are too many managers and clients to deal with.
- [ ] The information is difficult to understand for other stakeholders.
- [ ] I don't know what information to give to other stakeholders.
- [ ] Other stakeholders can't find the information or it is difficult to obtain the information.
- [ ] The information contains errors.
- [ ] The information given to the stakeholders is not what they need.
- [ ] Other (please describe)

22. Are there other problems in software engineering that you think are related to this concept of transparency?

- ○ Yes
- ○ No
- ○ Don't know

If yes, please describe the problems.

Transparency in software development

We define transparency in software engineering as:
Enabling stakeholders to answer their questions about the software project.

A stakeholder can be anyone involved in the software project, e.g. users, market analysts, software engineers.

We believe that in order to achieve this concept of transparency, the information presented in software projects should have the following attributes:

- ***Accessibility.*** Information is accessible when it can be obtained easily.
- ***Relevance.*** Information is relevant when it is appropriate to the expectations of the stakeholders.
- ***Understandability.*** Information is understandable when it can be perceived by any stakeholders with reasonable knowledge.

23. How important do you think the attributes that we describe are to this concept of transparency in software engineering?

	Very important	Important	Neutral	Not important	Not related to transparency
Accessibility	○	○	○	○	○
Relevance	○	○	○	○	○
Understandability	○	○	○	○	○

24. Are there other attributes that you think will be important to this concept of transparency?

- ○ Yes
- ○ No
- ○ Don't know

If yes, please describe the attributes.

Transparency in software development

25. In order to achieve this concept of transparency, what type of information in the software project do you think needs to be accessible, relevant and understandable? (Select the three most important)

- [ ] Business objectives
- [ ] User requirements
- [ ] System specification
- [ ] Software architecture
- [ ] Design rationale
- [ ] Bug report
- [ ] User manual
- [ ] Cost analysis
- [ ] Risk analysis
- [ ] Market research
- [ ] Other (please specify)

26. How effective do you think the following techniques will help in making information...to stakeholders?

a) Accessible. The information is accessible when stakeholders are able to obtain the information easily.
b) Relevant. The information is relevant when the information obtained is appropriate to the expectations of the stakeholders.
c) Understandable. The information is understandable when stakeholders, with reasonable knowledge, are able to perceive the information.

Use of prototypes.

	Very good	Good	Satisfactory	Poor	Very poor	N/A
a) Accessible	○	○	○	○	○	○
b) Relevant	○	○	○	○	○	○
c) Understandable	○	○	○	○	○	○

Use of diagrams, e.g. use case diagrams.

	Very good	Good	Satisfactory	Poor	Very poor	N/A
a) Accessible	○	○	○	○	○	○
b) Relevant	○	○	○	○	○	○
c) Understandable	○	○	○	○	○	○

Use of formal methods, e.g. formal notations such as Z.

	Very good	Good	Satisfactory	Poor	Very poor	N/A
a) Accessible	○	○	○	○	○	○
b) Relevant	○	○	○	○	○	○
c) Understandable	○	○	○	○	○	○

Have regular meetings/discussions with other stakeholders.

	Very good	Good	Satisfactory	Poor	Very poor	N/A
a) Accessible	○	○	○	○	○	○
b) Relevant	○	○	○	○	○	○
c) Understandable	○	○	○	○	○	○

Transparency in software development

Have a communication channel that allows for continuous feedback, e.g. use of emails.

	Very good	Good	Satisfactory	Poor	Very poor	N/A
a) Accessible	○	○	○	○	○	○
b) Relevant	○	○	○	○	○	○
c) Understandable	○	○	○	○	○	○

Other (please comment on what and how effective other techniques are).

Overall comments

27. Please comment on any other problems or concerns that you have regarding the concept of transparency.

28. Please comment on any other problems or concerns that you have in general regarding software engineering.

Transparency in software development

5. Thank you

Thank you for taking part in this research. We appreciate your time and value your response in the questionnaire.

D
Ethics Application for the Controlled Experiment

1. Completed Research Project Application Form to the University of Auckland Human Participants Ethics Committee.
2. Participant Information Sheet (Online Questionnaire).
3. Participant Information Sheet (Written Questionnaire).
4. Participants Consent Form (Written Questionnaire).
5. Assurance Letter from Department of Computer Science.
6. Poster.
7. Email Templates.

RESEARCH/COURSEWORK PROJECT APPLICATION FORM **Ref: ____________** ***(For office only)***

Prior to completing the application form, please check whether:
(i) exemption applies (refer the **Guiding Principles Section 3e**), or
(ii) the matter needs to be referred to a Regional Ethics Committee or a Multi-Centre Ethics Committee for approval (refer the **Guiding Principles Section 3c**).

The Guiding Principles for Research and Applicant's Manual can be found on The University of Auckland Human Participants Ethics Committee (UAHPEC) website.

Please note that the questions in the Research Project and Coursework application form are not exactly the same. Please refer to the Notes below.

Notes:

1. On all applications from students (including Doctoral, Masters and Honours students), the Principal Investigator should be the appropriate supervisor.
2. Questions prefixed with an **R** are applicable to a **Research Project** only.
3. Questions prefixed with a **C** are applicable to a **Coursework Application** only.
4. Questions without **R** or **C** are applicable to both Research Projects and Coursework Applications.
5. Questions prefixed with an asterisk (*****) are mandatory.
6. Where an email address is requested, the email address used must be from The University of Auckland.
7. Questions that are answered with a fixed set of choices (like **Yes** or **No**) have the possible answers separated with a slash (/). Please delete the incorrect answer.

Please contact the UAHPEC Ethics Administrator at 373 7599 extn: 87830/83761/83711 or e-mail: humanethics@auckland.ac.nz if there are any queries on these procedures.

GENERAL INFORMATION

* Is this a Research Project or Coursework Application? **Research**

SECTION A: PERSONNEL

***R**	A:1 **Principal Investigator**	Name:	Ewan Tempero
		Department:	Computer Science
		E-Mail Address:	e.tempero@cs.auckland.ac.nz
		I.D. Number:	
		Signature:	

	A:2 **Co-Investigator**	Name:	Clark Thomborson
		E-Mail Address:	c.thomborson@auckland.ac.nz
		I.D. Number:	
	A:3 **Student**	Name:	Yu-Cheng Tu
		E-Mail Address:	ytu001@aucklanduni.ac.nz
		I.D. Number:	3350030
		Signature:	
***C**	A:1 **Course Coordinator**	Name:	
		Department:	
		E-Mail Address:	
		I.D. Number:	
		Signature:	
***C**	A:2 **Course Administrator**	Name:	
		I.D. Number:	
	A:4 **Ethics Advisor**	Name:	
		E-Mail Address:	
		I.D. Number:	
		Signature:	

• A:5 **Maori Ethics Advisor** Name:

E-Mail Address:

I.D. Number:

Signature:

* A:6 **Head of School / Department**

Name: Gill Dobbie

E-Mail Address: gill@cs.auckland.ac.nz

I.D. Number:

Signature:

SECTION B: RESEARCH PROCEDURES

*R B:1 **Project Title**

Comparing the effectiveness of software document types in the presentation of functional requirements

*C B:1 **Paper Name & Number**

* B:2 **Aims/Objectives of Project**

The main objective of this research project is to study the effectiveness of two different types of software documents in presenting functional requirements of a software system. We would like to know how well the software documents are in helping people to find information and to understand the functionality of a software system. In this research project, we aim to ask software practitioners as well as tertiary students studying in areas related to software engineering to review software documents.

To achieve our objectives, we aim to answer the following research questions:

- Which software document enables people to find a particular functional requirement more easily?
- Which software document helps people to answer questions about the

functionality of a software system more correctly?

- Which software document takes less time for people to review the functionality of a software system?
- Can people find any inconsistencies or errors from the software documents?
- Can people find anything missing from the software documents?

Describe in plain language the purpose, hypothesis/research questions and objectives of the research in language that is comprehensible to lay people and free from jargon.

NOTE: All acronyms must be written out in full the first time they appear in the application, recruiting materials, Participant Information Sheet (PIS) and Consent Form (CF).

***R** B:3 **Summary of the Project (max 2000 characters)**

In our previous study, we asked people involved in software projects to identify any problems related to information gathering and communication that they have encountered in software projects. We also asked people about various definitions of transparency in different contexts as well as our proposed definition of transparency in software development. We found that people were familiar with the concept of transparency, but they were not so familiar with its use to software development. We believe that, if our definition is formalised in practice, software developers would be more successful in communicating with other stakeholders in a software project. The information provided by software developers would be more accessible, more relevant, and more understandable to stakeholders. Stakeholders would be able to assess the information provided by software developers more effectively.

To test our belief about transparency in software development, we aim to study the effectiveness of the software artefacts used in communicating with stakeholders during software development. In particular, we focus on two types of software documents used for conveying functional requirements of a software system to stakeholders. We would like to compare functional requirements written in *unstructured natural language* with *use case models*. We hypothesise that *use case models* are more transparent than *unstructured natural language*. Information presented in *use case models* would be more accessible, more relevant, and more understandable to stakeholders. We also hypothesise that *use case models* are more effective in helping stakeholders to review the functionality of a software system.

In this research project, we would like to ask software practitioners and tertiary students to review software documents in the form of either *unstructured natural language* or *use case models*. We aim to test our hypothesis about whether *use case models* are more transparent than *unstructured natural language* or not. We also aim to see which software document takes less time for software practitioners and tertiary students to find information about a software system as well as how well they answer questions about the functionality of a software system.

Please provide a detailed description of the project and its background which places the project in perspective and allows the Committee to assess its significance.

*R B:4 **Project Duration**

Start Date: 16/05/2012 End Date: 31/12/2012

* B:5 **Describe the study design**

This study is a controlled experiment, which involves participants to review software documents in the form of either unstructured natural language requirements or use case models. Participants will be asked to answer a set of questions using the software documents given to them. The study will consist of the following parts.

- **Demographics: In this section, participants will be asked a few general questions about their study if they are currently studying. They will be asked about the type of software models or modelling languages that they have learned during their study. For participants who are working in the software industry, they will be asked questions about their roles in software projects, years working in the software industry as well as their views on different types of software document and models.**
- **Part 1 (reviewing functionality of a software system): In this section, participants will be asked to read a software document and answer questions about the functionality of a software system described in the software document.**
- **Part 2 (overview of the software document): In this section, participants are asked to comment on the quality of the software documents. Participants will also be asked to comment on how well they think that they have answered The questions in part 1, and how they would improve the given software documents for presenting functional requirements of a software system.**

* B:6 **List all the methods used for obtaining information.**

We will use web-based questionnaires and written questionnaires for the controlled experiment. Written questionnaires will be used for participants who will be able to physically attend the experiment. If participants are unable to attend the experiment session, web-based questionnaire will be used.

Interviews: No

Note: If "yes", please attach the Interview Schedule when submitting your application.

Focus Groups: No

Note: If "yes", please attach the Focus Group Questions when submitting your application.

Questionnaires: Yes

Note: If "yes", please attach the Questionnaire when submitting your application.

Observations: No

Other: No

(If "yes" to Other, please explain)

* B:7 **Does the research involve processes that involve EEG, ECG, MRI, TMS, FMRI, EMG, radiation, invasive or surface recordings?** No

(If "yes", please explain)

* B:8 **Does the research involve processes that are potentially disadvantageous to a person or group (for example, the collection of information which may expose the person/group to discrimination)?** No

(If "yes", please explain)

* B:9 **Who will carry out the research procedures?**

The student (Yu-Cheng Tu).

If the research procedures will be carried out by a third party other than the researcher or co-investigators, please attach a copy of the confidentiality agreement when submitting your application.

* B:10a **Where will the research procedures take place?**

We will book a private room for participants who are taking part in the written questionnaires. For other participants, the research procedures will take place using web-based questionnaires.

Please attach the appropriate Request for Site Access and Consent Form when submitting your application, if necessary.

***R** B:10b **Will the research be conducted overseas?** No

(If "yes", please indicate which countries are involved.)

Please provide local contact details as well as those of contacts at the University – all of these should appear in the PIS.

***R** B:10c **If the study is based overseas, explain what special circumstances arise and how they will be dealt with. Include any special requirements of the country (e.g. research visa) and/or the community with which the research will be carried out.**

(If study is based overseas, please explain.)

Please also provide an undertaking to abide by any local laws relating to research, privacy and data collection.

***** B:11a **If a questionnaire is used, is the questionnaire web-based?** Yes

Note: If "yes", please indicate this on the PIS

***** B:11b **If a questionnaire is used, is it an anonymous questionnaire?** No

(If "yes", please explain (and indicate on the PIS) how anonymity will be preserved.)

***** B:12 **How much time will participants need to give to the research?**

(How many minutes/hours over how many weeks/months) **Approximately 1 hour.**

Please indicate this on the PIS.

***R** B:13 **Will information on the participants be obtained from third parties?** No

(If "yes", please explain)

Note: If Yes, please explain (and indicate on the PIS) & attach a copy of the Support Letter where necessary when submitting your application. For example: information is to be obtained from participant's employer, teacher, doctor, etc.

***R** B:14 **Will any identifiable information on the participants be given to third parties?**

No

(If "yes", please state who and explain)

Normally identifiable information or recorded interviews cannot be shared with third parties. If this is intended it must be clearly documented in the PIS for all concerned.

* B:15 **Does the research involve evaluation of University of Auckland services or organisational practices where information of a personal nature may be collected and where participants may be identified?** No

(If "yes", please explain and indicate this on the PIS)

* B:16 **Does the research involve a conflict of interest or the appearance of a conflict of interest for the researcher?** No

(If "yes", please explain and indicate this on the PIS)

***R** B:17 **Does the research involve matters of commercial sensitivity?** No

(If "yes", please explain and indicate this on the PIS)

***R** B:18 **Has the study design or the use of data been influenced by an organisation outside The University of Auckland?** No

(If "yes", please explain)

***R** B:19 **Are you intending to conduct the research in The University of Auckland class time?** No

Please attach the approval from the Course Coordinator when submitting your application.

* B:20 **Does the research involve deception of the participants, including concealment or covert observations?** No

(If "yes", please justify its use and describe the debriefing procedure on the PIS)

Please attach the debriefing sheet when submitting your application.

*R B:21 **Is there any koha, compensation or reimbursement of expenses to be made to participants?** Yes

(If "yes", please explain the level of payment and indicate in the PIS) **Participants who participated in the written questionnaire will be rewarded a movie voucher as a compensation of their time and effort at the end of the session. However, participants who answered the web-based questionnaire will not be rewarded. This is due to difficulties in locating where the participants are and identifying who have actually answered the web-based questionnaire.**

* B:22a **Is this an intervention study?** No

(If "yes", please explain and indicate this on the PIS)

* B:22b **Does this research involve potentially hazardous substances?** No

(If "yes", please explain and indicate this on the PIS)

*C B:23 **Will there be participants from outside this class?** Yes / No

(If "yes", please explain who they are and how much time will be required)

SECTION C: PARTICIPANTS

* C:1 **Who are the participants in the research?**

- Adults: **Yes**
- Own Colleagues: **No**
- Own Students: **Yes**
- Persons whose capacity to give informed consent (other than children) is compromised: **No**
- Persons who are in a dependent situation, such as people with a disability, residents of a hospital, nursing home or prison, or patients highly dependent on medical care: **No**
- Persons aged less than 16 years old where parental consent is being sought: **No**

- o Persons aged less than 16 years old where parental consent is NOT being sought: **No**

Note: If you answered "yes" to the question (above) on where parental consent is not sought for persons aged less than 16 years old, please indicate the age range of the persons below and explain in Section D2a & b

Less than 7 years old: **Yes / No**

Greater than 7 and less than 16 years old: **Yes / No**

Other: **Yes / No**

(If "yes" to Other, please explain)

* C:2 **How many organisations and departments within the organisations within or outside of the University of Auckland will participate in your project?**

5 – 30 organisations/departments within organisations.

If you have letters of support, please attach these when submitting your application.

*R C:3 **How many individual participants (research participants) will participate in your project?** _We wish to ask 30 – 100 individuals to participate in our research project._

* C:4 **How will you identify potential participants and by which method are participants invited to take part in the research?**

(Please explain) **Recruitment of participants will occur via email and advertisements. Email invitations will be sent to potential participants. Yellow pages and online directory will be used to identify potential software organisations. In addition, a mailing list of professional software engineers will be used with the permission of the mailing list owner and within the privacy declaration on the mailing list. Advertisements in the form of posters will be posted on notice boards within the University campus and with the permission of the University. People who are interested in the research project will be asked to reply to the email address provided in the email invitations and advertisements.**

Using a direct approach to recruit participants is not recommended. Please see the Applicant's Manual for further information. Please attach the advertisement, media release, or notice, etc and the letter of permission from the agency supplying them (if applicable) when submitting your application.

* C:5 **Who will make the initial approach to potential participants?**
Researcher and/or Other

(If "Researcher and/ or "Other", please specify and explain)
The researchers will make the initial approach by sending invitation for research participation to potential participants via email. Advertisements will be posted on the notice boards within the University campus by the student (Yu-Cheng Tu). The student will also ask the owner of the mailing list to forward the email for invitation to research participation.

* C:6 **Will access to participants be gained with consent of any organisation?** No

(If "yes", please explain)

*If the research is to be conducted in any organisation, such as a business, non-governmental organisation or school, a separate PIS needs to be provided for the Chief Executive Officer, Principal or the owner of the business (i.e. the effective employer) seeking permission to access the employees as participants. See **Applicant's Manual Sections 2c-iv**.*

* C:7 **Is there any special relationship between participants and researchers?** No

(If "yes", please explain)

It will not be appropriate, usually, for the researcher to recruit, members of their own family and friends as participants.

*R C:8 **Does the research involve University of Auckland staff or students where information of a personal nature may be collected and where participants may be identified?** No

(If "yes", please explain and indicate this on the PIS)

* C:9 **Does the research involve participants who are being asked to comment on employers?** No

(If "yes", please explain and indicate this on the PIS)

*R C:10 **Are there any potential participants who will be excluded?** No

(If "yes", please explain and state in here the criteria for excluding participants)

SECTION D: INFORMATION AND CONSENT

* D:1 **By whom and how will information about the research be given to participants?**

(Please explain)
The student (Yu-Cheng Tu) will give the information about the research with the PIS by email upon receiving response of interest from prospective participants.

For example: A copy of information to be given to prospective participants in the form of a PIS must be attached to this application, whether this is to be given verbally or in writing.

* D:2a **Will the participants have difficulty giving informed consent on their own behalf?**
No

Consider physical or mental condition, age, language, legal status, or other barriers.

* D:2b **If participants are not competent to give fully informed consent who will consent on their behalf?**

Parent or Guardian/Caregiver: **Yes / No**

Other: **Yes / No**

(If "Other", please specify)

* D:3a **If a questionnaire is used, will the participants have difficulty completing the questionnaire on their own behalf?** No

Note: If yes, please answer the next question. If no, please skip the next question.

Consider physical or mental condition, age, language, legal status, or other barriers.

* D:3b **If participants are not competent to complete the questionnaire, who will act on their behalf?**

Parent or Guardian/Caregiver: **Yes / No**

Other: **Yes / No**

(If "Other", please specify)

* D:4 **Does the research involve participants giving oral consent rather than written consent?** No

(If "yes", please explain and justify and indicate this on the PIS)

* D:5 **Does the research use previously collected information or biological samples for which there was no explicit consent?** No

(If "yes", please explain)

*R D:6 **Is access to the Consent Forms restricted to the Principal Investigator and/or the researcher?** Yes

(If "no", please explain and justify and indicate this on the PIS)

In general, the CF can only be accessed by the PI and the researcher.

* D:7 **Will Consent Forms be stored by the Principal Investigator, in a secure manner?**

Yes

(If "no", please explain and justify and indicate this on the PIS)

In general, the CF has to be stored in a locked cabinet on university premises.

*R D:8 **Are Consent Forms stored separately from data and kept for six years?**
No

(If "no", please explain and justify and indicate this on the PIS) **For the web-based questionnaires, participants will not be required to sign consent forms separately. Completed questionnaires will be treated as consent forms.**

In general, the CF has to be stored separately from other data for six years.

SECTION E: STORAGE AND USE OF RESULTS

* E:1 **Will the participants be audio-taped, video-taped, or recorded by any other electronic means such as Digital Voice Recorders?** No

(If "yes", please indicate the types of recordings)

Note: If "no", please skip question E2 (a-d).

If recording is essential to the research, it should be indicated as such in all relevant PISs. The CF should state, 'I understand that I will be recorded'.

If recording is optional, this should be explained in the PIS. The CF should state "I agree / do not agree to be recorded". It should also state that, 'Even if you agree to being recorded, you may choose to have the recorder turned off at any time'. The PIS to Chief Executive Officers, Principals, and Board of Trustees should state recordings will be made only with the agreement of those recorded.

* E:2a **Will the recordings be transcribed or translated?** No

Note: If "yes", please indicate this on the PIS & CF.

*Where any document is to be distributed to participants, it is to be provided for those participants in the language that will provide the most readily accessible presentation of adequate information. The UAHPEC requires English versions of documents to be submitted with an application. The UAHPEC does not require translations to be submitted with the application, but does expect to receive them after approval of the application and before they are used. For Languages other than English, see **Applicant's Manual Section 3j.***

* E:2b **Who will be transcribing the recordings?**
Researcher / Other / Researcher and/or Other

(If "Researcher and/or "Other", please explain in PIS & CF who will do the transcription (if not the researcher) & how confidentiality of information will be preserved. Please attach Confidentiality Agreement when submitting).

* E:2c **If recordings are made, will participants be offered the opportunity to edit the transcripts of the recordings?** Yes / No

(Please explain)

Only those who are recorded should be given the opportunity to review tapes or transcripts. Chief Executive Officers, for example, normally should not be given access to recordings made of their employees, nor to transcripts of these. If those who have been recorded are permitted to review tapes or transcripts, a clear *description should be provided in the PIS of the procedures for doing this. Where participants are asked to make a choice, this should be explained in the PIS and CF.*

* E:2d **Will participants be offered their tapes or digital files of their recording (or a copy thereof)?** Yes / No

(If yes, please explain)

Indicate in the PIS who will own the recorded data and how the data will be disposed of at the completion of the study. Options include, but are not limited to the participants retaining the recording, agreeing that the recording be destroyed, or consenting to its storage in a research archive.

If the data have not been publicly archived, which requires the participant's agreement, storage should be accessible by the researcher and supervisor only. Where participants are asked to make a choice, this should be explained in the PIS and CF.

* E:3 **For the questionnaire, is any coding scheme used to identify the respondent?** No

(If "yes", please explain)

Explain the coding procedure in the PIS. For example: Questionnaires are numbered 1-999 and a list is maintained to link participants with the questionnaire

* E:4a **Explain how and how long the data (including audio-tapes, video-tapes, digital voice recorder, and electronic data) will be stored.**

All data provided by the participants via web-based questionnaires will be stored in electronic devices within the University premises. Written questionnaires will be stored in a locked office within the University premises. All data will be used and stored for six years.

*Explain in the PIS and CF in what format data will be stored. The period data is to be kept will be commensurate to the scale of its research. For peer reviewed publication that might be further developed, the University expects six years. See **Applicant's Manual Section 2c-ii and 3n.***

* E:4b **Explain how data will be used.**

Data will be analysed and reported anonymously for the PhD research of the student. Findings will be reported in the student's PhD thesis and in publications.

Note: Please indicate this on the PIS.

*R E:4C **Explain how data will be destroyed.**

All electronic data will be deleted from all devices in a secure manner. All paper data will be destroyed using a paper shredder and disposed securely.

Please explain in the PIS & CF in what format, data will be subsequently destroyed.

* E:5 **Describe any arrangements to make results available to participants.**

The raw data will not be made available to the participants. A summary of the research findings will be made available online.

Researchers should be aware that there is an ethical dimension to the formulation and publication of results and loss of copyright. The researcher must remain sensitive to the uses to which the research findings may be put. Wherever possible, the findings should be conveyed in a comprehensible form to those who participated in the research. Explain this in the PIS.

***R** E:6a **Are you going to identify the research participants in any publication or report about the research?** No

Note: If "yes", the PIS must inform the participants, and this must be part of the consent obtained in the CF. If "no", please answer the next question.

***R** E:6b **Is there any possibility that individuals or groups could be identified in the final publication or report?** No

(If "yes", please explain here and describe in the PIS)

SECTION F: TREATY OF WAITANGI

* F:1 **Does the proposed research have impact on Māori persons as Māori?**
No

Note: If "yes", please answer the remaining questions in this section. If "no", please go straight to Section G.

* F:2 **Explain how the intended research process is consistent with the provisions of the Treaty of Waitangi.**

* F:3 **Identify the group(s) with whom consultation has taken place, describe the consultation process, and attach evidence of the support of the group(s) when submitting the application.**

* F:4 **Describe any on-going involvement the group(s) consulted has in the project.**

* F:5 **Describe how information will be disseminated to participants and the group consulted at the end of the project.**

* F:6 **List all the Māori methodology used for obtaining information.**

Predominant use of a kanohi ki te kanohi (face to face) approach when establishing networks, interacting and engaging with individuals and organizations: **Yes / No**

The use of karakia and appropriate protocols to conduct hui: **Yes / No**

The use of powhiri, whakatau and mihimihi processes: **Yes / No**

The use and promotion of te reo Maori: **Yes / No**

The use of protective mechanisms regarding cultural and intellectual property of participants: **Yes / No**

The use and significance of kai: **Yes / No**

The use and active practice of culturally appropriate processes wherever possible: **Yes / No**

Other: **Yes / No**

(If "Other", please explain)

SECTION G: OTHER CULTURAL ISSUES

* G:1 **Are there any aspects of the research that might raise any specific cultural issues?**
No

Note: If "yes", please answer the remaining questions in this section. If "no", please go straight to Section H.

* G:2 **What ethnic or cultural group(s) does the research involve?**

* G:3 **Identify the group(s) with whom consultation has taken place, describe the consultation process, and attach evidence of the support of the group(s) when submitting your application.**

* G:4 **Describe any on-going involvement the group(s) consulted has in the project.**

*R G:5 **Describe how information will be disseminated to participants and the group(s) consulted at the end of the project.**

Note: Please indicate this on the PIS and CF.

SECTION H: RISKS AND BENEFITS

*R H:1 **What are the possible benefits to research participants of taking part in the research?**

Participants who participate in the written questionnaire will be rewarded a movie voucher for their time and effort. However, there is no direct benefit to the participants taking the web-based questionnaire.

There will be benefits to student participants. Students will be able to know about how software documents may look like in the software industry. They will also have the opportunity to learn how functional requirements of a software system can be presented.

There will be benefits to the software engineering community. The result of this research will help researchers to test their hypotheses about transparency. Researchers will be able to identify attributes of software documents that will help stakeholders to find and understand information about a software system more effectively. The software industry will be able to articulate problems in software documents using the attributes. Software practitioners will be able to improve the quality of information provided to stakeholders based on the attributes identified in this research project.

* H:2 **Is the research likely to place the researcher at risk of harm?** No

(If "yes", please clearly identify/explain these risks here and in the PIS and CF)

* H:3 **Is the research likely to cause any possible harm to the participants, such as physical pain beyond mild discomfort, embarrassment, psychological or spiritual harm?** No

(If "yes", please clearly identify/explain these risks here and in the PIS and CF)

* H:4 **Does the research involve collection of information about illegal behaviour(s) which could place the researcher or participants at risk of criminal or civil liability or be damaging to their financial standing, employability, professional or personal relationships?** No

(If "yes", please clearly identify/explain these risks here and in the PIS and CF)

* H:5 **Is it possible that the research could give rise to incidental findings?** No

(If "yes", please explain how you will manage the situation)

Note: Clearly identify/explain these risks in the PIS and CF.

* H:6 **Describe what provisions are in place for the research participants should there be adverse consequences or physical or psychological risks.**

Participants will be asked to record the code on the software documents given to them in the questionnaire. The code is used for the researchers to identify which of the two software documents were being used to answer the questionnaire. Participants will not be identified with the code used.

Moreover, we will not ask for any information that can be used to identify the participants in the questionnaire. We will not collect any information that will identify other individuals or organisations that the participants work for. However, it is possible that the participants might accidentally reveal any personal information. To reduce the likelihood, participants will be made aware of the risks before consenting to participate in the research. We will also ask the participants not to give any identifying information in their responses. We will remove any

personal or organisational information disclosed. We will not make the data publicly available on the Internet. Access to the data will be limited to the researchers of this research project. In addition, the data collected will be analysed and reported anonymously.

Note: Please explain this in the PIS and CF.

SECTION I: HUMAN REMAINS, TISSUE AND BODY FLUIDS

* I:1 **Does the research involve use of human blood, body fluids, or tissue samples?**
No

Note: If "no", please go to Section J. If "yes", please explain in the PIS. Provide a copy of the information to be given to the Transplant Coordinator (if necessary), and state the information that the Transplant Coordinator will provide to those giving consent. Complete the remaining questions in this section.

* I:2 **Are these samples obtained from persons involved in research or will the tissue be obtained from a tissue bank?**

* I:3 **Is the tissue imported or taken in New Zealand?** Imported / NZ

(If "imported", please indicate the country of origin)

Note: If ethics approval is obtained from the country of origin, please attach the approval when submitting your application.

* I:4 **Describe how the sample / specimen is taken.**

* I:5a **Is blood being collected?** Yes / No

Note: If "yes", please indicate this on the PIS and answer the next 4 questions. If "no", please skip the next 4 questions.

* I:5b **What is the volume at each collection?**

* I:5c **How frequent are the collections?**

* I:5d **Who is collecting it?**

* I:5e **Is the collector trained in phlebotomy?** Yes / No

(If "no", please explain)

* I:6a **Will the sample / specimen be retained for possible future use?** Yes / No

(If "yes", please explain and state this in the PIS and CF)

Note: If "yes", please answer the next 2 questions. If "no", please skip the next 2 questions.

* I:6b **Where will the material be stored?**

* I:6c **How long will it be stored for?**

* I:7a **Will material remain after the research process?** Yes / No

(If "yes", please explain and state this in the PIS and CF)

Note: If "yes", please answer the next 2 questions. If "no", please skip the next 2 questions

* I:7b **How will material be disposed of?**

* I:7c **Will material be disposed of in consultation with relevant cultural groups?** Yes / No

(If "yes", please explain and state this in the PIS and CF)

SECTION J: CLINICAL TRIALS

***R** J:1 **Is the research considered a clinical trial?** No

Note: If "yes", please include the declaration of the trials in the PIS under Compensation" and attach Form A or Form B when submitting your application and answer the remaining questions in this section. If "no", please go straight to Section K.

UAPHEC adopts the definition of clinical trial of the World Health Organisation and New Zealand Ministry of Health. That definition is 'a clinical trial is any research study that prospectively assigns human participants or groups of humans to one or more health-related interventions to evaluate the effects on health outcomes'. See ***Applicant's Manual Section 5d*** *for the declaration of the trials and Forms A and B.*

***R** J:2 **Is this project initiated by a Pharmaceutical Company?** Yes / No

Note: If "yes", please attach the letter from the Pharmaceutical Company when submitting your application.

***R** J:3 **Are there other NZ or International Centres involved?** Yes / No

Note: If "yes", please attach the support letter when submitting your application.

***R** J:4 **Is there a clear statement about indemnity?** Yes / No

(If "no", please explain)

Note: If "yes", please attach a copy of the indemnity when submitting your application.

***R** J:5 **Is Standing Committee on Therapeutic Trials (SCOTT) approval required?** Yes / No

Note: If "yes", please attach a copy of the SCOTT approval when submitting your application.

***R** J:6 **Is National Radiation Laboratory (NRL) approval required?** Yes / No

Note: If "yes", please attach a copy of the NRL approval when submitting your application.

***R** J:7 **Is Gene Therapy Advisory Committee on Assisted Human Reproduction (NACHDSE) approval required?** Yes / No

Note: If "yes", please attach a copy of the NACHDSE approval when submitting your application.

SECTION K: FUNDING

***R** K:1 **Have you applied for, or received funding for this project?** No

Note: If "yes", please answer the remaining questions in this section. If "no", please go straight to Section L.

***R** K:2 **From which funding institution?**

***R** K:3a **Is this a UniServices project?** Yes / No

***R** K:3b **Is this a Research contract?** Yes / No

***R** K:3c **Is this a Commercial or consulting contract?** Yes / No

***R** K:4 **Contract reference number**

***R** K:5 **Do you see any conflict of interest between the interests of the researcher, the participants or the funding body?** Yes / No

(If "yes", please explain)

SECTION L: OTHER INFORMATION

***** L:1 **Have you made any other related applications?** Yes

(If "yes", please provide Approval Reference Number) **2010/484**

***** L:2 **Is there any relevant information from past applications or interaction with UAHPEC?**
No

(If "yes", please indicate it here and attach the relevant information when submitting your application)

* L:3 **Please provide a summary of all the ethical issues arising from this project and explain how they are to be resolved.**
(For example: confidentiality, anonymity, informed consent, participant's rights to withdraw, conflict of interest, etc.)

Confidentiality: We are unable to make guarantees to complete confidentiality of the responses made from the participants. This is because we have little control of data privacy on the Internet when using web-based questionnaires. There is some small risk of exposing data collected from the participants to other parties on the Internet. To mitigate this risk, all data collected will not be made available on the Internet. We will also store the written questionnaires in a locked office within the university premises. The consent forms will also be stored in a locked office, but separately from the questionnaire. Access to all the data collected for this research project will be limited to the researchers of this research project.

Anonymity: In this research project, it is not possible to collect the written questionnaires anonymously. For written questionnaires, participants' handwriting might be recognisable by the student (Yu-Cheng Tu). To minimise this risk, the consent forms will be distributed and collected separately from the questionnaire. The student (Yu-Cheng Tu) will distribute the consent forms at the beginning of the session. Once the participants have agreed to participate and signed the consent forms, the student will collect the forms and distribute the questionnaire. When collecting the consent forms, the student will ask the participants to put the forms facing down on the table so the student will not see their handwriting. The student will not be able to associate the names with participants' handwriting. The student will put the forms in an envelope and delivered to the PI unopened. The consent forms will be stored separately from the questionnaire. Handwriting from the consent forms will not be used for comparing with the questionnaire.

Moreover, we will not identify individual participants in their responses. We will not ask for any information that directly reveals participants' identities in the questionnaire. Participants will be warned not to provide any personal information in their responses. We will also remove any identifying information disclosed from the responses. We will analyse and report the information provided by the participants anonymously.

Rights to withdraw: Participant's rights to withdraw data from the research would not be possible. This is because we will not be able to identify individual participants and their responses. Participants will be made aware that they are unable to withdraw data after submitting the questionnaire in the PIS. However, participants are entitled to withdraw from involvement in the research at any time before submitting the questionnaire.

Informed consent: Participants will not be required to sign consent forms for the web-based questionnaires. However, we will include a consent page at the beginning of the questionnaire that enables participants to indicate if they understand what is involved in the research. We will also note to the participants that by submitting the questionnaire indicates that they agree to participate in the research.

Conflict of interest: There is some small likelihood that our participants are students of the principal investigator and co-investigator. Participants will only be contacted by the student

(Yu-Cheng Tu). Moreover, the experiment sessions will be conducted by the student (Yu-Cheng Tu) only. The principal investigator and co-investigator will not be involved in the sessions. We will also emphasise in the PIS that the information collected for this research project will not affect students' grades. When recruiting potential student participants, we will rely on advertising and we will not make any personal approaches to students.

UAHPEC expects applicants to identify the ethical issues in the project and explain in the documentation how they have been resolved. The application will not be considered if this is not answered adequately. A "Not applicable" response is not acceptable.

SECTION M: APPLICATION CHECKLIST

* Have you attached the Participant Information Sheet? (See Applicant's Manual Sections 2b-ii, 2c and 5a for explanation and sample): **Yes**

* Have you attached the Consent Form? (See Applicant's Manual Sections 2b-iii, 2d and 5b for explanation and sample): **Yes**

* Have you attached the advertisement? : **Yes**

* Have you attached the questionnaire? : **Yes**

* Have you attached the list of interview questions? : **No**

* Have you attached the confidentiality agreement? (See Applicant's Manual Sections 2b-vii and 5c for explanation and sample) : **No**

* Have you attached any other supporting documents (for example: approval from Course Coordinator, debriefing sheet)? : **No**

* Have you completed the Application Checklist (Preliminary Assessment)? : **Yes**

APPLICATION CHECKLIST (Please delete whichever is not applicable)

Preliminary Assessment				
A.	**Risk of Harm**			
	1.	Does the research involve situations in which the researcher may be at risk of harm?		NO
	2.	Does the research involve the use of any method, whether anonymous or not, which might reasonably be expected to cause discomfort, pain, embarrassment, psychological or spiritual harm to the participants?		NO
	3.	Does the research involve processes that are potentially disadvantageous to a person or group, such as the collection of information which may expose the person/group to discrimination?		NO
	4.	Does the research involve collection of information about illegal behaviour(s) which could place the researcher or participants at risk of criminal or civil liability or be damaging to their financial standing, employability, professional or personal relationships?		NO
*	5.	Does the research involve any form of physically invasive procedure on participants, such as the collection of blood, body fluids, tissue samples, DNA, human tissue from a tissue bank, exercise or dietary regimes or physical examination?		NO
*	6.	Does the research involve any intervention administered to the participant, such as drugs, medicine (other than in the course of standard medical procedure), placebo, environmental conditions, food/drink?		NO
*	7	Does the research involve processes that involve EEG, ECG, MRI, TMS, FMRI, EMG, radiation, invasive or surface recordings?		NO
*	8.	Is the research considered a clinical trial?		NO
	9.	Does the research involve physical pain beyond mild discomfort?		NO

B.	**Informed and Voluntary Consent**			
	1.	Does the research involve participants giving oral consent rather than written consent? (If participants are anonymous the response is "No").		NO
	2.	Does the research involve participation of children (seven years old or younger)?		NO
	3.	Does the research involve participation of children under sixteen years of age where parental consent is not being sought?		NO
	4.	Does the research involve participants who are in a dependent situation, such as people with a disability, residents of a hospital, nursing home or prison, or patients highly dependent on medical care?		NO
*	5.	Does the research involve participants who are being asked to comment on employers?		NO
	6.	Does the research involve participants (other than children) whose capacity to give informed consent is in doubt?		NO
	7.	Does the research use previously collected information or biological samples for which there was no explicit consent?		NO
C.	**Research conducted overseas**			
	1.	Will the research be conducted overseas?		NO
D.	**Privacy and confidentiality issues**			
	1.	Does the research involve evaluation of University of Auckland services or organisational practices where information of a personal nature may be collected and where participants may be identified?		NO
	2.	Does the research involve University of Auckland staff or students where information of a personal nature may be collected and where participants may be identified?		NO
	3.	Does the research involve matters of commercial sensitivity?		NO
	4.	Does the research involve Focus Groups?		NO
E.	**Deception**			
	1.	Does the research involve deception of the participants, including concealment or covert observations?		NO
F.	**Conflict of interest**			
*	1.	Does the research involve a conflict of interest or the appearance of a conflict of interest for the researcher (for example, where the researcher is also the lecturer/teacher/treatment provider/colleague or employer of the participants, or where there is a power relationship between researcher and participants)?	YES	
G.	**Cultural sensitivity**			
	1.	Does the research have impact on Maori?		NO
	2.	Does the research raise any specific ethnic or cultural issues not relating to Maori?		NO
H.	**Requirements imposed from outside The University of Auckland**			
	1.	Does the research involve a requirement imposed by an organisation outside The University of Auckland?		NO

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Department of Computer Science
The University of Auckland
Private Bag 92019
Auckland, New Zealand
Phone: +64 9 373 7599 ext 85857

Participant Information Sheet

Project title: Comparing the Effectiveness of Software Document Types in the Presentation of Functional Requirements (Online Questionnaire)

Researchers: Yu-Cheng Tu / Associate Professor Ewan Tempero / Professor Clark Thomborson

To: The Participant

This research project is being undertaken by Yu-Cheng Tu, a PhD student studying in Software Engineering. This research is a part of a PhD degree at the Department of Electrical and Computing Engineering, University of Auckland. The purpose of this research is to study the effectiveness of software documents in presenting functional requirements of a software system. Participants have been selected by responding to our email invitations or advertisements and agreeing to participate in this research.

Our research involves the use of an online questionnaire, which will take approximately 1 hour to complete. The questionnaire will consist of the following parts.

- **Demographics (5 minutes):** In this section, you will be asked a few general questions about your study if you are currently a tertiary student. If you are working in the software industry, you will be asked about your roles in software projects as well as your experience with different types of software documents and models.
- **Part 1. Reviewing functionality of a software system (40 minutes):** In this section, you will be asked to review a software document (attached in the email) that describes the functional requirements of a software system. You will need to answer questions about the software system based on the information provided in the document. Please do not spend more than 40 minutes on this part. You are not required to go through everything in the software document to answer the questions.
- **Part 2. Overview of the software document (15 minutes):** In this section, you will be asked some questions about the quality of the software document given to you and how you would improve the document for presenting functional requirements of a software system.

The data that you provide will be used for the PhD research of the student (Yu-Cheng Tu). Data will also be used to report findings of this research in conference papers and journal articles. Findings of this research will be reported in the student’s PhD thesis. Any identifying information such as your name and IP address will not be recorded in the data. All information that you provide in the questionnaire will remain anonymous. Data will not be made publicly available on the Internet. They will only be available to the researchers of this research project, and will be stored securely in electronic devices within the university premises for a period of six years. After that period, the information will be destroyed from all devices in a secure manner.

Participation in this study is voluntary. You can be assured that neither your grades nor academic relationships with the Department of Computer Science at The University of Auckland or any member of the staff will be affected by either refusal or agreement to participate in this study. This assurance is provided by the Head of Department of Computer Science. If you decide to participate in this research, you have the right to withdraw from participation at any time before the point of submitting the questionnaire. However, you are unable to withdraw data provided by you from the research after the point of submitting the questionnaire. This is because the data will not contain any information that will be identified as belonging to any particular participants. The information that you provide will be analysed and reported anonymously.

If you are willing to participate, please complete the online questionnaire at the link provided in the email. You will not be asked to sign a consent form. However, there will be an electronic consent at the beginning of the online questionnaire. Please note that by submitting the online questionnaire indicates that you agree to take part in this research.

A summary of the research findings will be made available online at http://www.cs.auckland.ac.nz/research/groups/ssg/homepages/yu-cheng/expSummary.html after the completion of this research project. If you have any questions about the research, please contact us. Contact details are provided below.

Contacts
Professor Gill Dobbie (Head of Department, Department of Computer Science)
Phone: +64 9 373 7599 ext 83949
Email: gill@cs.auckland.ac.nz

Associate Professor Ewan Tempero (Supervisor, Department of Computer Science)
Phone: +64 9 373 7599 ext 83765
Email: e.tempero@cs.auckland.ac.nz

Professor Clark Thomborson (Supervisor, Department of Computer Science)
Phone: +64 9 373 7599 ext 85753
Email: cthombor@cs.auckland.ac.nz

Yu-Cheng Tu (PhD student, Department of Electrical and Computing Engineering)
Phone: +64 21 0471916
Email: ytu001@aucklanduni.ac.nz

For any queries regarding ethical concerns you may contact the Chair, The University of Auckland Human Participants Ethics committee, The University of Auckland, Research Office, Private Bag 92019, Auckland 1142. Telephone 09 373-7599 extn. 83711. Email: humanethics@auckland.ac.nz.

APPROVED BY THE UNIVERSITY OF AUCKLAND HUMAN PARTICIPANTS ETHICS COMMITTEE ON 23 May 2012 FOR (3) years, Reference Number 8118

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Department of Computer Science
The University of Auckland
Private Bag 92019
Auckland, New Zealand
Phone: +64 9 373 7599 ext 85857

Participant Information Sheet

Project title: Comparing the Effectiveness of Software Document Types in the Presentation of Functional Requirements (Written Questionnaire)

Researchers: Yu-Cheng Tu / Associate Professor Ewan Tempero / Professor Clark Thomborson

To: The Participant

This research project is being undertaken by Yu-Cheng Tu, a PhD student studying in Software Engineering. This research is a part of a PhD degree at the Department of Electrical and Computing Engineering, University of Auckland. The purpose of this research is to study the effectiveness of software documents in presenting functional requirements of a software system. Participants have been selected by responding to our email invitations or advertisements and agreeing to participate in this research.

Our research involves the use of a written questionnaire, which will take approximately 1 hour to complete. You will be informed about the time and the location where the experiment session will be held in the email. The student (Yu-Cheng Tu) will be present throughout the experiment session. At the beginning of the experiment session, you will be given and asked to sign a consent form if you agree to take part in this study. To minimise the possibility of recognising your handwriting by the student (Yu-Cheng Tu), please place the consent form facing down on the table. After the consent form has been collected from you, you will be given a software document and a questionnaire. The questionnaire will consist of the following parts.

- **Demographics (5 minutes):** In this section, you will be asked a few general questions about your study if you are currently a tertiary student. If you are working in the software industry, you will be asked about your roles in software projects as well as your experience with different types of software documents and models.
- **Part 1. Reviewing functionality of a software system (40 minutes):** In this section, you will be asked to review a software document that describes the functional requirements of a software system. You will need to answer questions about the software system based on the information provided in the document. Please do not spend more than 40 minutes on this part. You are not required to go through everything in the software document to answer the questions.
- **Part 2. Overview of the software document (15 minutes):** In this section, you will be asked some questions about the quality of the software document and how you would improve the document for presenting functional requirements of a software system.

At the end of the experiment session or when you have completed the questionnaire, please return the written questionnaire to the student (Yu-Cheng Tu).

The data that you provide will be used for the PhD research of the student (Yu-Cheng Tu). Data will also be used to report findings of this research in conference papers and journal articles. Findings of this research will be reported in the student's PhD thesis. All information that you provide in the questionnaire will remain anonymous. Data will not be made publicly available on the Internet. They will only be available to the researchers of this research project, and will be stored in a locked office within the university premises for a period of six years. After that period, the information will be destroyed using a paper shredder and disposed securely.

Participation in this study is voluntary. You can be assured that neither your grades nor academic relationships with the Department of Computer Science at The University of Auckland or any member of the staff will be affected by either refusal or agreement to participate in this study. This assurance is provided by the Head of Department of Computer Science. If you decide to participate in this research, you have the right to withdraw from participation at any time before the point of submitting the questionnaire. However, you are unable to withdraw data provided by you from the research after the point of submitting the questionnaire. The consent form will be stored separately and will not be used to identify your handwriting. The consent form will not be associated with the questionnaire. The questionnaire will not contain any information that will be identified as belonging to any particular participants. The information that you provide will be analysed and reported anonymously.

At the end of the experiment session, you will be rewarded a movie voucher as a compensation for your time and effort in completing the questionnaire. A summary of the research findings will be made available online at http://www.cs.auckland.ac.nz/research/groups/ssg/homepages/yu-cheng/expSummary.html after the completion of this research project. If you have any questions about the research, please contact us. Contact details are provided below.

Contacts

Professor Gill Dobbie (Head of Department, Department of Computer Science)
Phone: +64 9 373 7599 ext 83949
Email: gill@cs.auckland.ac.nz

Associate Professor Ewan Tempero (Supervisor, Department of Computer Science)
Phone: +64 9 373 7599 ext 83765
Email: e.tempero@cs.auckland.ac.nz

Professor Clark Thomborson (Supervisor, Department of Computer Science)
Phone: +64 9 373 7599 ext 85753
Email: cthombor@cs.auckland.ac.nz

Yu-Cheng Tu (PhD student, Department of Electrical and Computing Engineering)
Phone: +64 21 0471916
Email: ytu001@aucklanduni.ac.nz

For any queries regarding ethical concerns you may contact the Chair, The University of Auckland Human Participants Ethics committee, The University of Auckland, Research Office, Private Bag 92019, Auckland 1142. Telephone 09 373-7599 extn. 83711. Email: humanethics@auckland.ac.nz.

APPROVED BY THE UNIVERSITY OF AUCKLAND HUMAN PARTICIPANTS ETHICS COMMITTEE ON 23 May 2012 FOR (3) years, Reference Number 8118

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Department of Computer Science
The University of Auckland
Private Bag 92019
Auckland, New Zealand
Phone: +64 9 373 7599 ext 85857

Participants Consent Form

THIS FORM WILL BE HELD FOR A PERIOD OF 6 YEARS

Project title: Comparing the Effectiveness of Requirements Documents in the Presentation of Functional Requirements (Written Questionnaire)

Researchers: Yu-Cheng Tu / Associate Professor Ewan Tempero / Professor Clark Thomborson

I have read the Participant Information Sheet, have understood the nature of the research and why I have been selected. I have had the opportunity to ask questions and have them answered to my satisfaction.

- I understand that this consent form will be stored separately from the questionnaire for a period of 6 years before it is destroyed.
- I understand that this consent form will not be used to associate my handwriting with the questionnaire.
- I agree to take part in this research.
- I understand that participation in this research project is voluntary.
- I understand that the Head of Department of Computer Science has provided signed assurance that neither my grades nor academic relationship with the Department of Computer Science at The University of Auckland or any member of the staff will be affected by either refusal or agreement to participate in this research project.
- I understand that I am free to withdraw participation at any time before the point of submitting the questionnaire.
- I understand that I will not be able to withdraw any data provided by me after the point of submitting the questionnaire.
- I understand that the questionnaire will take approximately 1 hour.
- I understand that I will receive a movie voucher as a compensation for my time and effort in completing the questionnaire.
- I understand that data will be stored and used for the PhD research of the student (Yu-Cheng Tu) for 6 years, after which they will be destroyed.

Name ______________________________

Signature ______________________________ Date ____________________

APPROVED BY THE UNIVERSITY OF AUCKLAND HUMAN PARTICIPANTS ETHICS COMMITTEE ON 23 May 2012 FOR (3) years, Reference Number 8118

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Department of Computer Science
The University of Auckland
Private Bag 92019
Auckland, New Zealand
Phone: +64 9 373 7599 ext 85857

16 May 2012

Assurance from Department of Computer Science

Project title: Comparing the Effectiveness of Requirements Documents in the Presentation of Functional Requirements. UAHPEC Reference number 8118

Researchers: Yu-Cheng Tu / Associate Professor Ewan Tempero / Professor Clark Thomborson

Dear Participant,

I assure you that neither your grades nor academic relationship with the Department of Computer Science at The University of Auckland or any member of the staff will be affected by either refusal or agreement to participate in the research project stated above.

G. C Dobbie

Professor Gill Dobbie
Head of Department
Department of Computer Science

EARN A MOVIE VOUCHER

IN A SOFTWARE DOCUMENT REVIEW STUDY!

Are you a tertiary student studying in Software Engineering, Computer Science, Information Technology or any other related areas?

A PhD student needs you to take part in a study!

The study is about investigating how effective different types of software documents are in presenting functional requirements to different people. It will take approximately 1 hour. You will be asked to review a software document and answer a questionnaire. You will receive a movie voucher at the end of the session.

If you are interested to participate, contact:

Yu-Cheng Tu ytu001@aucklanduni.ac.nz

PhD Student
University of Auckland

For a Limited Time Only!

APPROVED BY THE UNIVERSITY OF AUCKLAND HUMAN PARTICIPANTS ETHICS COMMITTEE ON 23 May 2012 for (3) years, Reference Number 8118

Email invitation to participate in the research.

Subject: Invitation to participate in a PhD research project at the University of Auckland

Hi,

My name is Yu-Cheng Tu, and I am a PhD student researching in Software Engineering at the University of Auckland. I am conducting a study to investigate how effective different types of software documents are in presenting functional requirements to different people. I would like to invite anyone who is either:

- involved in software development (e.g. software developer, requirements engineer, project manager, client of a software project), or
- tertiary students studying in Software Engineering, Computer Science, Information Technology, or any other related areas to take part in my research.

The research involves the use of a questionnaire (either online or written), which will take approximately 1 hour to complete. You will be given a software document that describes the functionality of a software system. In the questionnaire, you will be asked some questions about the software document.

Your participation in this research is voluntary. You may choose not to participate. However, your participation and feedback will be of great value to my research. The results will help my research in identifying attributes of software documents that will help stakeholders to find and understand information a software system more effectively. These attributes will be useful for software practitioners to improve the quality of information provided to stakeholders during software development.

If you are interested to participate in my research, please contact me at ytu001@aucklanduni.ac.nz before 2012. You may choose to complete the questionnaire online or during one of our experiment sessions. If you would like to participate in the experiment session, please also let me know the times that you will be available.

If you have any questions about this research, please feel free to contact me. Thank you for considering this request.

Sincerely,
Yu-Cheng Tu

PhD Student
University of Auckland
Supervisors: Assoc. Prof. Ewan Tempero, Prof. Clark Thomborson (Department of Computer Science)

Email template for replying to prospective participants (online questionnaire)

Hi (participant's name),

Thank you for your interest. Here is a short description about the research procedures:

The questionnaire will consist of a demographic section and two main parts. In the demographic section, you will be asked some general questions about your study and what software models you have learned during your study (if you are a student). If you are working in the software industry, you will have questions about your roles in software projects and how well you think different types of software documents are. In the first part of the questionnaire, you will be asked to answer questions about a software system based on the information provided in the software document attached to this email. In the second part of the questionnaire, you will be asked some questions about the quality of the software document and how you would improve the software document.

Attached is a copy of the Participant Information Sheet, which describes this research in detail. Please read it carefully. You will also find attached a copy of the software document that describes the functionality of a software system. If you are willing to participate, please complete the online questionnaire at http://www.surveymonkey.com/s/5GJ3CJT.

Thank you for taking part in this research project. If you have any questions, please feel free to contact me.

Sincerely,
Yu-Cheng Tu

PhD Student
University of Auckland
Supervisors: Assoc. Prof. Ewan Tempero, Prof. Clark Thomborson (Department of Computer Science)

Email template for replying to prospective participants (written questionnaire)

Hi (participant's name),

Thank you for your interest. Here is a short description about the research procedures:

At the beginning of the experiment session, you will be given a software document that describes the functionality of a software system and a questionnaire for you to complete. The questionnaire will consist of a demographic section and two main parts. In the demographic section, you will be asked some general questions about your study and what software models you have learned during your study (if you are a student). If you are working in the software industry, you will have questions about your roles in software projects and how well you think different types of software documents are. In the first part of the questionnaire, you will be asked to answer questions about a software system based on the information provided in the software document given to you. In the second part of the questionnaire, you will be asked some questions about the quality of the software document and how you would improve the software document. At the end of the session, you will receive a movie voucher as a compensation for your time and effort.

Attached is a copy of the Participant Information Sheet, which describes this research in detail. Please read it carefully. If you are willing to participate, please let me know when you are available. I will arrange the experiment session at a time that is convenient for you.

Thank you for taking part in this research project. If you have any questions, please feel free to contact me.

Sincerely,
Yu-Cheng Tu

PhD Student
University of Auckland
Supervisors: Assoc. Prof. Ewan Tempero, Prof. Clark Thomborson (Department of Computer Science)

E

Questionnaire for the Controlled Experiment

- Demographics (industry).
- Demographics (student).
- Part 1 Reviewing Functionality of a Software System.
- Part 2 Overview of the Software Document.

Comparing the Effectiveness of Software Document Types in the Presentation of Functional Requirements – Questionnaire

Thank you for taking part in this research project. Please try to answer every question in this questionnaire. You may choose not to answer any of the optional questions (marked with Optional) that you think will take more than 10 minutes to complete.

Demographics (5 minutes)

1. What aspects of the software project are you involved in? (Select all that apply)
   - ☐ Requirements
   - ☐ Design
   - ☐ Development
   - ☐ Testing
   - ☐ Maintenance
   - ☐ Project management
   - ☐ Quality management
   - ☐ Other (please specify)

2. What roles do you generally have in a software project? (Select all that apply)
   - ☐ Requirements engineer
   - ☐ Developer
   - ☐ Architect
   - ☐ Project manager
   - ☐ User / user representative
   - ☐ Client
   - ☐ Regulator
   - ☐ Other (please specify)

3. How many years have you been working in the software industry?

   ○ 0 - 4 years ○ 5 - 9 years ○ 10 - 14 years ○ 15 - 19 years ○ 20+ years

4. How do you usually get to know the requirements for the software product?

   (Select all that apply)

   ☐ I consult comprehensive documentation.

   ☐ I consult informal documentation.

   ☐ I learn about the requirements by informal discussions with other members of my organisation.

   ☐ I learn about the requirements by informal discussions with clients.

   ☐ I learn about the requirements at formal meetings with clients and/or other members of my organisation.

   ☐ Other (please specify)

5. What type of documents or models do you usually use or receive from the ways you use in Q4 (to know about the requirements for the software product)? How effective do you think the documents or models are in helping you to understand the functional requirements of the software product?

   a) Requirements written in natural language (the requirements document does not follow any specific formats or standards such as ISO documentation standards).

How effective is this?	Comments
○ Very good ○ Good ○ Satisfactory ○ Poor, I have to guess the requirements ○ Poor, but I have to use it ○ I never receive or use this for understanding the requirements	

b) Requirements written in structured natural language (the requirements document follows a specific format or standard such as ISO documentation standards, IEEE standards or templates provided by your organisation).

How effective is this?	Comments
◯ Very good ◯ Good ◯ Satisfactory ◯ Poor, I have to guess the requirements ◯ Poor, but I have to use it ◯ I never receive or use this for understanding the requirements	

c) Requirements written in formal notations such as Z.

How effective is this?	Comments
◯ Very good ◯ Good ◯ Satisfactory ◯ Poor, I have to guess the requirements ◯ Poor, but I have to use it ◯ I never receive or use this for understanding the requirements	

d) Use case models.

How effective is this?	Comments
◯ Very good ◯ Good ◯ Satisfactory ◯ Poor, I have to guess the requirements ◯ Poor, but I have to use it ◯ I never receive or use this for understanding the requirements	

e) Entity-relationship diagrams.

How effective is this?	Comments
◯ Very good ◯ Good ◯ Satisfactory ◯ Poor, I have to guess the requirements ◯ Poor, but I have to use it ◯ I never receive or use this for understanding the requirements	

f) Architecture/design documents.

How effective is this?	Comments
○ Very good ○ Good ○ Satisfactory ○ Poor, I have to guess the requirements ○ Poor, but I have to use it ○ I never receive or use this for understanding the requirements	

g) User manuals.

How effective is this?	Comments
○ Very good ○ Good ○ Satisfactory ○ Poor, I have to guess the requirements ○ Poor, but I have to use it ○ I never receive or use this for understanding the requirements	

h) Informal diagrams such as rich picture, storyboards, spray diagram.

How effective is this?	Comments
○ Very good ○ Good ○ Satisfactory ○ Poor, I have to guess the requirements ○ Poor, but I have to use it ○ I never receive or use this for understanding the requirements	

i) Other (please specify)

6. Optional: What type of documents or models do you prefer to use for understanding functional requirements of a software product. Please also comment on why you prefer using such documents or models.

End of demographic questions

Comparing the Effectiveness of Software Document Types in the Presentation of Functional Requirements – Questionnaire

Thank you for taking part in this research project. Please try to answer every question in this questionnaire. You may choose not to answer any of the optional questions (marked with Optional) that you think will take more than 10 minutes to complete.

Demographics (5 minutes)

1. Which institution are you currently studying at?

2. What degree and major are you studying?

3. Which year are you in your study?

   ○ year 1 ○ year 2 ○ year 3 ○ year 4 ○ year 5+

4. Did you learn any software models or modelling methods during your study? (Select all that apply)

   ☐ State machines
   ☐ Use Case Models
   ☐ Data flow diagrams
   ☐ Activity diagrams
   ☐ Entity-relationship diagrams
   ☐ Unified Modelling Language (UML)
   ☐ I didn't learn any software models or modelling methods
   ☐ Other (please specify)

5. Do you have any working experience in the software industry? If yes, please also answer Q6 – Q9.

   ○ Yes ○ No

6. How long have you been working in the software industry?

   ○ 0 - 4 years ○ 5 - 9 years ○ 10 - 14 years ○ 15 - 19 years ○ 20+ years

7. How do you usually get to know the requirements for the software product?

   (Select all that apply)

   ☐ I consult comprehensive documentation.
   ☐ I consult informal documentation.
   ☐ I learn about the requirements by informal discussions with other members of my organisation.
   ☐ I learn about the requirements by informal discussions with clients.
   ☐ I learn about the requirements at formal meetings with clients and/or other members of my organisation.
   ☐ Other (please specify)

8. What type of documents or models do you usually use or receive from the ways you use in Q7 (to know about the requirements for the software product)? How effective do you think the documents or models are in helping you to understand the functional requirements of the software product?

   a) Requirements written in natural language (the requirements document does not follow any specific formats or standards such as ISO documentation standards).

How effective is this?	Comments
○ Very good ○ Good ○ Satisfactory ○ Poor, I have to guess the requirements ○ Poor, but I have to use it ○ I never receive or use this for understanding the requirements	

b) Requirements written in structured natural language (the requirements document follows a specific format or standard such as ISO documentation standards, IEEE standards or templates provided by your organisation).

How effective is this?	Comments
○ Very good ○ Good ○ Satisfactory ○ Poor, I have to guess the requirements ○ Poor, but I have to use it ○ I never receive or use this for understanding the requirements	

c) Requirements written in formal notations such as Z.

How effective is this?	Comments
○ Very good ○ Good ○ Satisfactory ○ Poor, I have to guess the requirements ○ Poor, but I have to use it ○ I never receive or use this for understanding the requirements	

d) Use case models.

How effective is this?	Comments
○ Very good ○ Good ○ Satisfactory ○ Poor, I have to guess the requirements ○ Poor, but I have to use it ○ I never receive or use this for understanding the requirements	

e) Entity-relationship diagrams.

How effective is this?	Comments
○ Very good ○ Good ○ Satisfactory ○ Poor, I have to guess the requirements ○ Poor, but I have to use it ○ I never receive or use this for understanding the requirements	

f) Architecture/design documents.

How effective is this?	Comments
○ Very good ○ Good ○ Satisfactory ○ Poor, I have to guess the requirements ○ Poor, but I have to use it ○ I never receive or use this for understanding the requirements	

g) User manuals.

How effective is this?	Comments
○ Very good ○ Good ○ Satisfactory ○ Poor, I have to guess the requirements ○ Poor, but I have to use it ○ I never receive or use this for understanding the requirements	

h) Informal diagrams such as rich picture, storyboards, spray diagram.

How effective is this?	Comments
○ Very good ○ Good ○ Satisfactory ○ Poor, I have to guess the requirements ○ Poor, but I have to use it ○ I never receive or use this for understanding the requirements	

i) Other (please specify)

9. Optional: What type of documents or models do you prefer to use for understanding functional requirements of a software product. Please also comment on why you prefer using such documents or models.

End of demographic questions

Part 1. Reviewing Functionality of a Software System (40 minutes)

Please spend no more than 40 minutes on this section. Please answer the following questions based on the information presented in the document as best as you can. If you cannot answer a question or if you feel it is taking too long to answer a question, please write down the problem in one or two sentences. For example, "I can't find the answer from the given document after spending 10 minutes", "I don't understand the question", etc.

1. Please select the code written on the top left of the document given to you.

   ○ ReqSpec2012

   ○ UCM2012

2. Please write down the start time for answering this part of the questionnaire.

3. What is the name of the system that is the primary web authentication system for the University?

4. What are the requirements for handling applications submitted in hard copies? (Please spend no more than 10 minutes on this question)

   Please describe where and how you found information about this functionality. Please note down the page numbers and section headings that you have looked for finding this functionality. E.g. I first read the table of contents. Then I went through Section 2 and 3. The requirement is on pg. 2, Section 2 Solution Overview.

5. How are hard-copy applications being processed?

6. Are the NCEA exam results displayed to the applicant? Please also note down the page numbers and/or section headings where you found the answer.

7. Who can run reports from the UAM system? Please also note down the page numbers and/or section headings where you found the answer.

8. Please write down the finish time for answering this part of the questionnaire.

End of part 1 questions

Part 2. Overview of the Software Document (15 minutes)

1. Optional: Did you find any duplicated or redundant information in the given document?

   ○ Yes ○ No

   Please write down what information was duplicated or redundant in the document.

2. Optional: Did you find any inconsistencies or errors in the given document? E.g. errors in the terminology used in the document.

   ○ Yes ○ No

   Please describe what inconsistencies or errors you have found in the document.

3. Optional: Did you find any information missing for describing the functionality of the software system?

   ○ Yes ○ No

   Please describe what information you think is missing in the document.

4. Did you have to go through different parts of the document (e.g. going to different sub sections) in order to answer Q6 about NCEA exam results in part 1?

	Comments
○ Yes ○ No	

5. How well do you think the given document is in helping you to...

			Comments
a)	Identify the information that you may need to answer the questions in part 1?	○ Very good ○ Good ○ Satisfactory ○ Poor ○ Very poor	
b)	Read only the relevant information that you need to answer each question in part 1?	○ Very good ○ Good ○ Satisfactory ○ Poor ○ Very poor	
c)	Understand the functionality of the software system?	○ Very good ○ Good ○ Satisfactory ○ Poor ○ Very poor	

6. How well do you think that...

			Comments
a)	You have understood the information provided in the given document?	○ Very good ○ Good ○ Satisfactory ○ Poor ○ Very poor	
b)	You have answered the questions in part 1 correctly?	○ Very good ○ Good ○ Satisfactory ○ Poor ○ Very poor	

7. Optional: If you ran out of time to answer questions in Part 1, what were the problems you encountered?

8. Optional: Please comment on whether you liked or didn't like the given software document. How would you improve the document for presenting the functionality of the software system?

9. Optional: Please comment on any problems or concerns that you have in general regarding software artefacts produced (e.g. software documents, class diagrams, test suite, etc.), or communication with other stakeholders during the life cycle of a software product.

End of the questionnaire

F
UAM IMS Requirements Specification

Experiment Code: ReqSpec2012

UAM/IMS Integration

UAM IMS Requirements Specification

Document Ref	**Title**		**External reference**
UAM-SPC-001	UAM IMS Requirements Specification		
Version Date	**Version #**	**Document Owner**	**Current Status**
20 March 2012	4.0		Draft

Table of Contents

1 Summary

1.1 Background

The University Accommodation Management system, UAM, is a specialised student housing solution. Applicants for student accommodation register by completing an online application via a link from the University website.

The Identity Management System (IMS) was introduced to the University in 2008 and is the University's system of record for identity information and primary web authentication. There is currently no automated integration between UAM and the IMS.

In order to bring some efficiency and provide a better user experience for students it is desirable to integrate the UAM system with the IMS – ensuring that a student only needs to register and update personal data in one system. Additionally, it would be beneficial to pass various affiliation information from other University systems to UAM to keep them informed of any student’s status changes.

1.2 Scope

The following items are considered to be in scope;

- Integration with the IMS for the Accommodation Application process to eliminate the need for students/applicants to register and enter personal details in more than one system.
- UAM Integration with IMS for any changes or updates to personal details for accommodation applicants or current students who have previously registered online.
- The integration of relevant Affiliations a person has with the University to UAM as they are up updated in the various downstream systems.

1.3 Dependencies

This project is dependent upon:

- ITS resource being available to do the development
- The accommodation package (UAM) being able to integrate suitably with the IMS

1.4 References

None

1.5 Glossary of terms

Term	Meaning
UAM	University Accommodation Management (UAM) system is the University's specialised student housing solution.
SAP	The Student Administration Platform.
IMS	The Identity Management System (IMS) is the University's master repository for Personal Data. It stores and maintains details of all persons that the University has a relationship with, including students, staff, visitors, alumni and contractors.
AfUE	Application for University Entrance. This is an application for recording applications for admission to programmes of study at the University. It interfaces with the IMS for the Personal and Contact details of the applicants.
ULN	University Login Name – assigned to persons in the IMS when an Identity becomes resolved.

2 Solution Overview

When a prospective or current student wants to submit an online Application for Accommodation at the University they will use the links currently provided on various web pages, e.g. the University Home page 'Accommodation' link, or via the Application for University Entrance (Accommodation Services link).

The solution for accommodation follows the pattern already established for similar web forms like the Application for University Entrance. Applicants visiting the Application for Accommodation for the first time will be required to register themselves in the IMS and provide all the necessary personal data. The applicant will then be transferred back to the accommodation form where their personal data will be displayed 'read only'. A link back to the IMS will be provided in order that applicants can update personal data at any time.

Additionally, it would be useful for Accommodation staff to have basic student-related information provide to UAM from other systems, to assist them in processing an applicant's request for accommodation.

3 Functional Specification

3.1 Process models

3.1.1 Account Registration

Under an Integrated systems approach, the 'Account Login' section on the University Accommodation Home page will direct the user to the IMS for Sign in. For someone who has previously registered with the university, they can use their University ID (7 character number), ULN (if they are already a student) or the personal email address they used to create their account. If they are new to the University then they will be taken to the IMS registration screen (for Accommodation applicants). They will be required to enter the following;

1. Email Address
2. First Name
3. Last Name
4. Password

Following completion of this form and acceptance of the terms and conditions they will be sent a confirmation email. They will be required to complete the verification by clicking on the 'Complete Your Registration' link and populating the following information (some of which is compulsory).

1. Title
2. Middle Names
3. Preferred Name
4. Previous Name
5. Mobile Phone Number (required)
6. Home Phone Number (required)
7. Correspondence Address (required)
8. Gender (required)
9. Date of Birth (required)
10. Citizenship (required)

11. Residency (if not an NZ citizen)
12. Ethnicity (required)
13. Emergency Contact details (required)
14. National Student Number (NSN)

Once this page is complete the applicant can return to the Accommodation portal – Home Page and lodge an application.

Note: Where an UAM administrator enters an application on behalf of an applicant, they will need the ability to create the identity in the IMS first. Then they will enter the IMS ID number into UAM manually and push the person message from the IMS in order to populate the required personal data fields. This process for gathering personal data would also be the same for non-student accommodation residents, who are currently entered into UAM through the Administration pages.

3.1.2 Completing an Accommodation Application – IMS Identity Data

Once an Applicant has either logged in or registered (in the IMS) and completed their verification they can begin a new Application, by selecting the 'My Application' link from the home page – as they do today. The UAM system will recognize them as being logged in, receiving their credentials from the IMS, using the 7 character unique University ID as the link between the two systems.

The Select Application page of the UAM system will look no different to how it is currently and once an applicant selects their application type and clicks on 'Save and Continue' - they will be transferred to the 'Personal Details' page. This page will be modified from what they see today, as all fields (not just 'Family Name' and 'First Name') will be display only. A link will exist on the page to 'Update Personal Details', which will transfer the user to the IMS (in a new browser window) where they can maintain their own data. When they return to the 'Personal Details' page their applicant information will be updated with whatever was saved in the IMS. This will be a near real-time update.

Similarly, with the 'Contact Details' page in the Accommodation Application – all fields will be read only with three separate buttons to link to different parts of the IMS for adding and/or updating Addresses, Phone & Email and Emergency Contacts details. A new browser window will be opened in the IMS on the appropriate page – and when saved will push a near-real time message to the Accommodation system to update anything that has changed.

3.1.3 Synchronising Affiliation data

Some affiliations (which represent the relationships a person has with the University) are displayed in the IMS. These are maintained in a number of University systems and sent via messaging to University Login Management system and LDAP (which is where the IMS is reading these from). When these affiliations are updated (or when a user is initially sent to the Accommodation system) these will be sent via message to the UAM database (for internal use only) using an Application Programming Interface (API) - supplied by UAM. This will enable administrators to have up to date information relating to an applicant's (or current resident's) status with the University. The Affiliations of interest to UAM are;

a. Applicant
b. Undergraduate Student
c. Postgraduate Student
d. Doctoral Student

e. Alumni

3.1.4 Other Student Related data

NCEA Test Results

When an applicant selects the 'School Leaver' option on the UAM application form they will be prompted for their National Student Number (NSN) if it is not already stored against their IMS identity. Note: NSN will be included in the personal data collection once an applicant verifies their email address, but will be optional at this stage. When an applicant proceeds to select their Accommodation Application type, and they select Option1 (School Leaver), a check on the database will be made to determine if their NSN number has been collected. If not, it will prompt them for it. When it is entered, or on confirmation that it has been entered, a web service will pull all relevant NCEA test data from the SAP database. Note: at this stage CIE and IB results sent from the Ministry of Education do not contain the NSN number so it will not be possible to collect these results automatically using the NSN number.

Scholarship Information

As part of the Education question in the Accommodation Application, information is gathered on whether an applicant has applied for or intends to apply for a scholarship. If an applicant has previously applied the information relating to this application is currently stored in SAP (though in the future this will be in Scholarship Management) and could be retrieved by a web service and posted directly to the UAM database via an API. The current scholarship question will be left in place, for cases where applicants intend to apply or have applied for a scholarship that is not centrally managed, i.e. some faculties manage their own scholarships.

Photos

All students are required to have an ID card photo entered in the university ID card system. A message is published from this system every time a new photo is added for a student. There is a requirement for the UAM system to subscribe to this message where a student is a current accommodation applicant or resident. This will update the UAM database directly via an API.

3.2 Business Rules / Regulatory Requirements

Type of Rule	**Identifier**	**Rule details**
Regulation		All requirements of the Public Records Act 2004 must be observed and adhered to.
Regulation		All requirements of the Privacy Act 1993 must be observed and adhered to.
Regulation		All requirements of the University's Employment Code - Access to Personal Information policy must be observed and adhered to.

3.3 Assumptions

	Assumption
1.	The Identity Management System will be the master for all personal data and will be responsible for sending any changes relating to people that the Accommodation system is interested in as they are updated or added.

3.4 Functional Requirements

Requirement #	Requirement Description	Navigation	Dependencies / Traceability
1	An Accommodation Applicant should only be required to register once with the University in order to apply for Admission to a Programme of Study and to stay in a University Residence. A single ID should be used to access both Applications and eventually be used as their Student ID once they are accepted into a Programme.		
2	Personal and Contact Details required for the Accommodation Application should be created and maintained in the IMS. A user should be able to access the IMS to amend these details directly from an Application within UAM.		
3	The UAM system should be updated automatically when a person (who is either an Applicant, a current University Accommodation resident or who has been offered a place in a University residence) changes or updates their personal or contact details in the IMS.		
4	The UAM system should be updated automatically when a person's (who is either an Applicant, a current University Accommodation resident or who has been offered a place in a University residence) affiliations with the university change.		

5	A report is required to identify persons with current accommodation applications who are not current students and whose applications for admission have been declined.		
6	A report is required to identify current University residents who are no longer active students (or were never enrolled).		
7	Photos from a student's id card are required to be sent to UAM when they are added or updated in the ID card system.		
8	Secondary School details and NCEA results (from Year 12 onwards) should be sourced from SAP data (and populated in the UAM database) if this available at the time an applicant completes the form – and we have received their NSN.		
9	Scholarship information should be pre-populated in the UAM database, from SAP, if it exists at the time an applicant completes the form.		
10	Hard Copy Applications – for those students that submit a hard-copy application, Accommodation staff require the ability to create user accounts in the IMS (on their behalf). They can then manually enter the IMS ID into UAM and then push the relevant person message from the IMS.		
11	Non-Student Applications – the business require the ability to process applications for non-students also. These persons will still require the same IMS registration (entered manually by UAM staff) and be pushed from the IMS in the same way as is the case with manually entered student applicants.		

3.4.1 Account Registration (online Student Applicants)

HOME

The University Accommodation Portal

Applications for 2012 Accommodation:

- Apply for Full Academic year and Semester One from 1 August 2011
- Apply for Semester Two from 1 August 2012

Welcome to the Accommodation Portal. You must register with the University to receive your login details. You need these details to log in, to return to the application, to check the status of the progress of your application and to correspond with Accommodation Services. Note also that this ID will be used for any University Applications for University Entrance and will eventually serve as your student ID once you have been accepted for a Programme of study.

From the Portal you can:

- Apply for a place in a University residence
- Check your application status
- Reapply as a returning resident

Applications

To apply if you are:

- **A new applicant, not yet registered with the University:** Please click on the 'Sign up for a new account' link below
- **A new applicant, already registered with the University:** Please sign in using the link below
- **A returning resident:** Please sign in using the link below

Application Progress Status

To check the status of the progress of your application, please sign in using the link below

Sign in

University ID, ULN or Email

Password

Sign in

Sign up for a new account

I cannot access my account

From the 'Sign up for a new account' link:

Register for a new account

Email address

First name

Last name

Choose a password

Confirm password

☐ I accept the terms and conditions, and disclaimer.

Register

Step 1

Complete and submit this registration form, using your personal email address. Please provide an alternative to your school email if possible. We will use this address to correspond with you.

Step 2

Select a secure password that is at least 8 characters in length.

Step 3

You will receive an email. Please use this to activate your account so it can be used to sign into the Accommodation Portal.

Clicking on the register Button will invoke the following message;

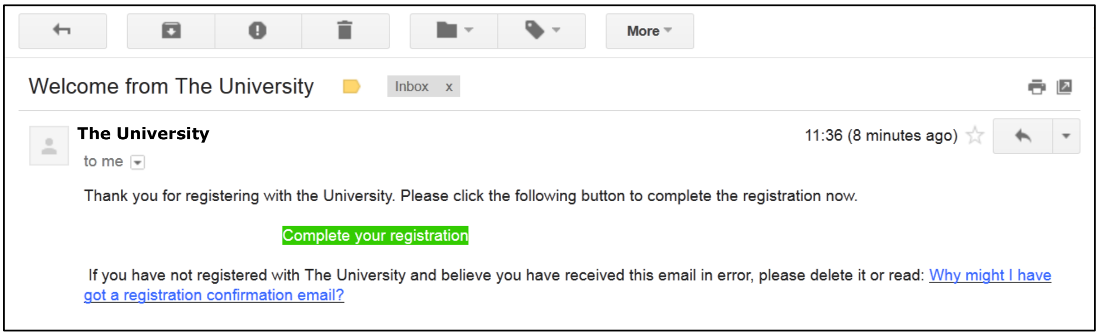
Validate your email address

We have sent an email to the following address. Please click on the link within the email to validate your address.

- xxxxxx@xyz.com

Continue

Confirmation Email from IMS



Clicking on the 'Complete your registration' link will bring up the following message;

Email address successfully validated

Thank you for validating your email address with the University.

If you were logging into an application, you may now close this window and continue to the application.

Validate your email address

We have sent an email to the following address. Please click on the link within the email to validate your address.

- xxxxxx@xyz.com ✓

Continue

Student (and potential student) Application
Clicking on the 'Continue' button opens up the Personal and Contact Details Page (tbd)

Application for Accommodation

*Required fields

Your name

Full legal name
Important: Please ensure the name reflects the legal name on passport or birth certificate

Title ☐
*First name ☐
Middle names ☐
*Last name ☐

Preferred name
Use this section to indicate other names

Do you have a preferred name that is different from your full legal name? ☐Yes ☐No
Do you have a previous or maiden name? ☐Yes ☐No

Your contact details

*Home phone ☐ Preferred Contact number
*Mobile phone ☐ Preferred Contact number
*Mailing address

Start typing your address. If you have an overseas address select Enter Overseas Address, or if you can't find your NZ address, select Manually enter a NZ address.

Enter overseas address or **Manually enter a NZ Address**

*Home address ☐ Same as Mailing address

Your demographics

*Gender ☐Male ☐Female
*Date of birth
*Citizenship
*Ethnicity

Emergency Contact

*Contact Name
*Relationship
Email
Home phone *
Mobile phone
Work phone
Address

Your National Student Number (NSN)

If you are applying as a New Zealand School Leaver then please provide your NSN

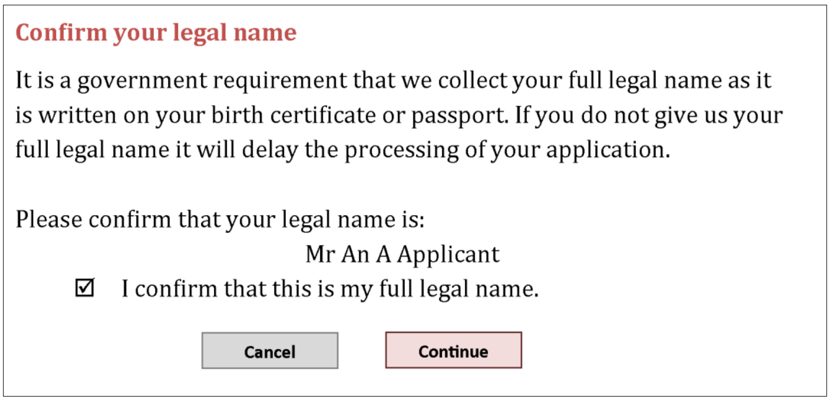
NSNs are the unique numbers used to identify students on the New Zealand National Student Index

Next

If a person selects a citizenship other than 'New Zealand', 'Australia' or 'Cook Islands' they are presented with this question (per the current AfUE)

*Citizenship	
*Are you a permanent resident of New Zealand?	☐Yes ☐No

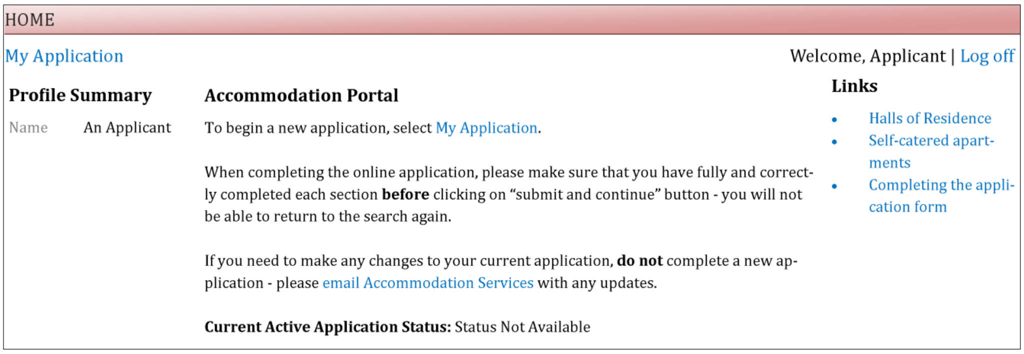
Click on Next once all Fields are populated and receive the following Confirmation message;



Clicking on Continue will take you into the 'Welcome' page (with Profile Summary in top left hand corner) of the Accommodation Application.

3.4.2 Completing an Accommodation Application

Once signed in or immediately after completing the Registration steps in Section 3.4.1 an applicant will be directed to the 'Welcome' page.


HOME

My Application

Welcome, Applicant | Log off

Profile Summary

Name An Applicant

Accommodation Portal

To begin a new application, select My Application.

When completing the online application, please make sure that you have fully and correctly completed each section **before** clicking on "submit and continue" button - you will not be able to return to the search again.

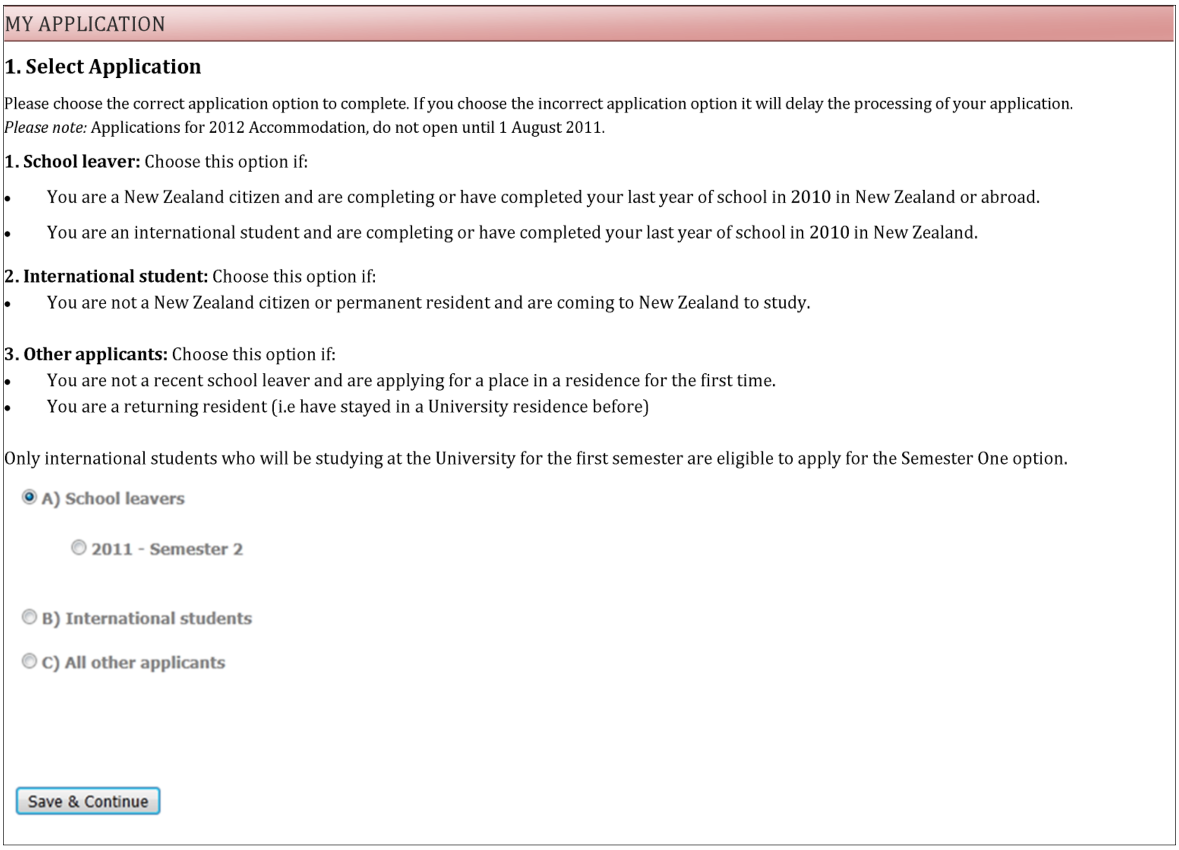
If you need to make any changes to your current application, **do not** complete a new application - please email Accommodation Services with any updates.

Current Active Application Status: Status Not Available

Links

- Halls of Residence
- Self-catered apartments
- Completing the application form

Using the 'My Application' link starts the Application process.


MY APPLICATION

1. Select Application

Please choose the correct application option to complete. If you choose the incorrect application option it will delay the processing of your application.
Please note: Applications for 2012 Accommodation, do not open until 1 August 2011.

1. School leaver: Choose this option if:

- You are a New Zealand citizen and are completing or have completed your last year of school in 2010 in New Zealand or abroad.
- You are an international student and are completing or have completed your last year of school in 2010 in New Zealand.

2. International student: Choose this option if:

- You are not a New Zealand citizen or permanent resident and are coming to New Zealand to study.

3. Other applicants: Choose this option if:

- You are not a recent school leaver and are applying for a place in a residence for the first time.
- You are a returning resident (i.e have stayed in a University residence before)

Only international students who will be studying at the University for the first semester are eligible to apply for the Semester One option.

A) School leavers

2011 - Semester 2

B) International students

C) All other applicants

Save & Continue

After selecting the type of applicant you are, and selecting 'Save & Continue' you are presented with your personal details. Note: the Personal and Contact Details pages are the same regardless of the Application type, e.g. School Leaver, International Student, or Other Applicants.

For Type 1 Applications (School Leavers), the following additional question should be asked, if the NSN is not already in the IMS for the applicant;

Your National Student Number (NSN)
If you are applying as a New Zealand School Leaver then please provide your NSN

NSNs are the unique numbers used to identify students on the New Zealand National Student Index

Next

This will enable the retrieval of any stored NCEA exam results in SAP (e.g. year 12 NCEA) for Applicants. Given the sensitive nature of this data the results should be only populated in the database tables and not displayed to the applicant.

MY APPLICATION

My Application Menu

Apply for **2012 - Semester 1**
A) **School leavers**

1. Personal Details
2. Contact Details
3. Education - Secondary School and Tertiary
4. Criminal Record
5. Residential Preference
6. Special Health and Other Requirements
7. Referees
8. Personal Profiles
9. Declaration
10. Application Status

1. Personal Details

Update Personal Details

Applicant Information

Last Name	Scott
First Name	Adam
Middle Name	Peter
Preferred Name	Peter Scott
University ID	1234567
Gender	Male
Date of Birth	04 May 1984
Citizenship	New Zealand
Residency	New Zealand Permanent Resident
Ethnicity	NZ Maori

Confirm & Continue

This page would be pre-populated with data from the IMS. Click on Update personal details to go to the IMS and change or add data (see below).

Clicking on the 'Confirm & Continue' button will take the user to the Contact Details page.


Personal details
Addresses
Email & Phone
Emergency Contacts
Integration
Affiliations
Audit
Legal name
Adam Peter Scott
Preferred name
Pete Scott
Update name
Demographics
Gender
Male
Date of birth
4 May 1984
Citizenship
New Zealand
Residency
You are a citizen of New Zealand.
Ethnicities
NZ Maori
Ethnic Group
Maori
Update demographics
Identifiers
Update identifiers
Status
Update status

MY APPLICATION

My Application Menu

Apply for **2012 - Semester 1**
A) School leavers

1. Personal Details
2. Contact Details
3. Education - Secondary School and Tertiary
4. Criminal Record
5. Residential Preference
6. Special Health and Other Requirements
7. Referees
8. Personal Profiles
9. Declaration
10. Application Status

2. Contact Details

Your Mailing Address for Correspondence & Residential Contract

Add / Update Address Details

Name of Correspondence Contact	Adam Scott
Street Address	10 Queen Street
Suburb	Auckland Central
City / Town	Auckland
Region	
Country	New Zealand
Postal Code	0530

Your Permanent Home Address

Street Address	23a Church Road
Suburb	Richmond
City / Town	London
Region	Surrey
Country	United Kingdom
Postal Code	TW10 23T

Your Email Addresses and Phone Numbers

Add / Update Email & Phone Details

Mobile Phone Number	012 3456789	☑ preferred
Home Telephone Number	0012 345 6789012	
Correspondence Email	apscott@gmail.com	

Your Emergency\Next-of-Kin Contact Details

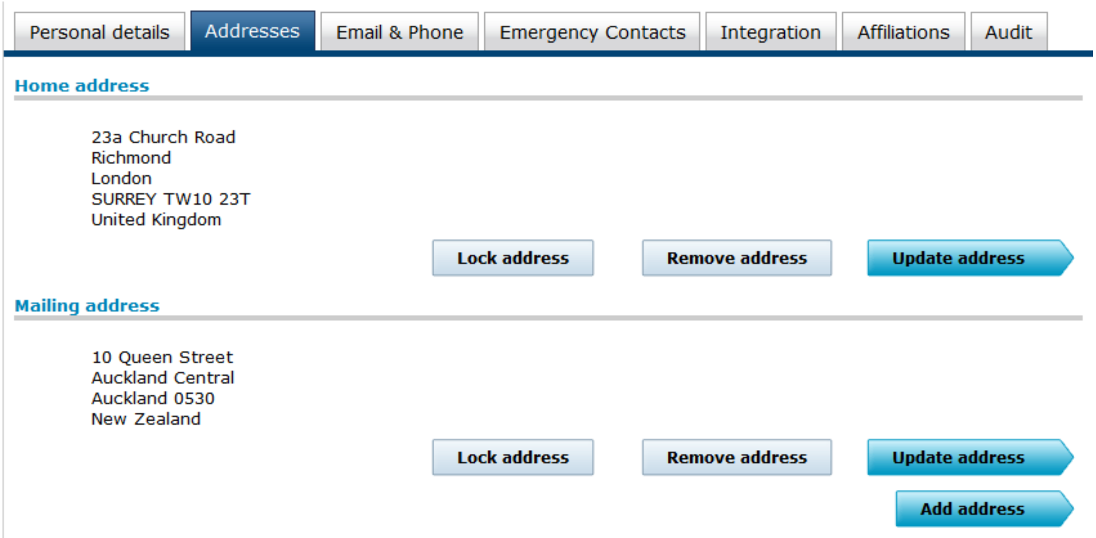
Add / Update Emergency Contact Details

Name of Correspondence Contact	Mary Scott
Relationship to you	Parent
Street Address	23a Church Road
Suburb	Richmond
City / Town	London
Region	Surrey
Country	United Kingdom
Postal Code	TW10 23T
Telephone Number	00442075689644
Mobile Phone	00440204567912
Correspondence Email	mscott@gmail.com

Confirm & Continue

The Add/Update IMS buttons will take the applicant to the following IMS pages to maintain their person data;

Add/Update Address Details – IMS


Personal details | Addresses | Email & Phone | Emergency Contacts | Integration | Affiliations | Audit

Home address

23a Church Road
Richmond
London
SURREY TW10 23T
United Kingdom

Lock address | Remove address | Update address

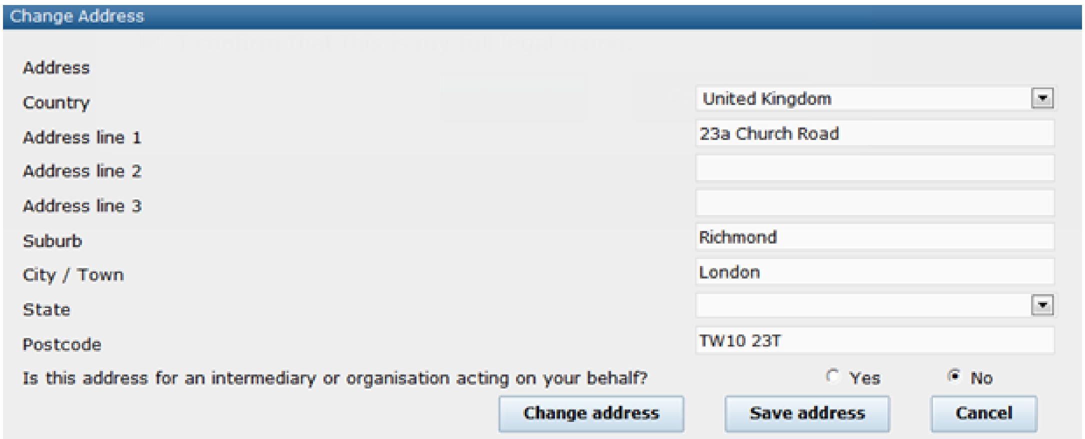
Mailing address

10 Queen Street
Auckland Central
Auckland 0530
New Zealand

Lock address | Remove address | Update address

Add address

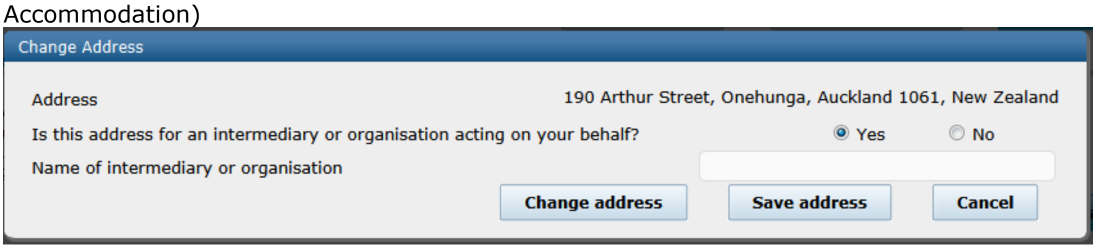
Click on Update Address


Change Address

Address	
Country	United Kingdom
Address line 1	23a Church Road
Address line 2	
Address line 3	
Suburb	Richmond
City / Town	London
State	
Postcode	TW10 23T
Is this address for an intermediary or organisation acting on your behalf?	Yes / No

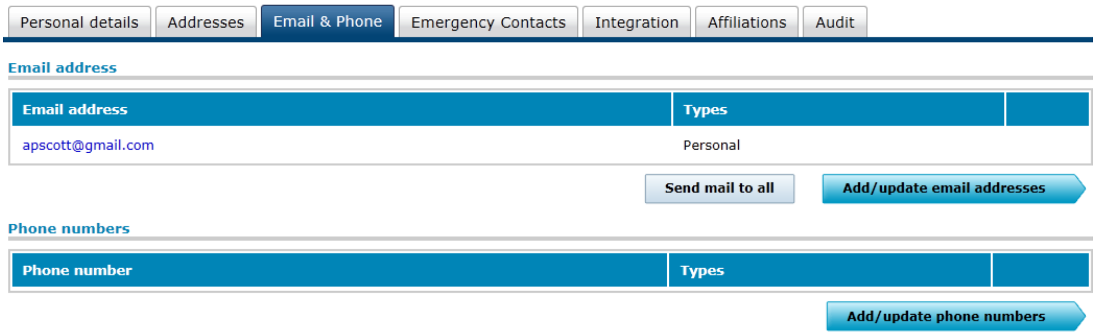
Change address | Save address | Cancel

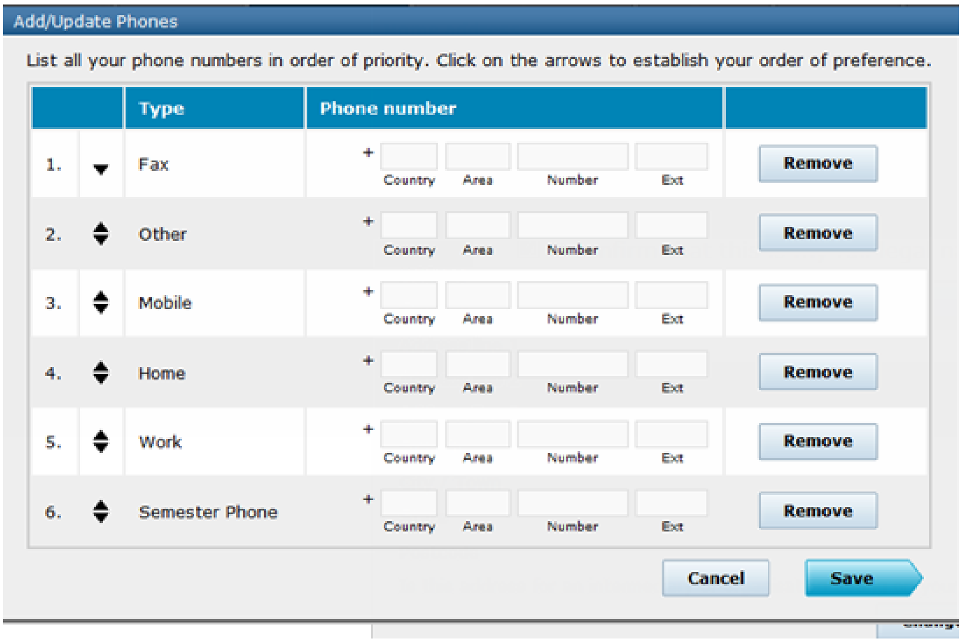
Note: When Entering an Address you can enter a Contact Name (required by Accommodation)

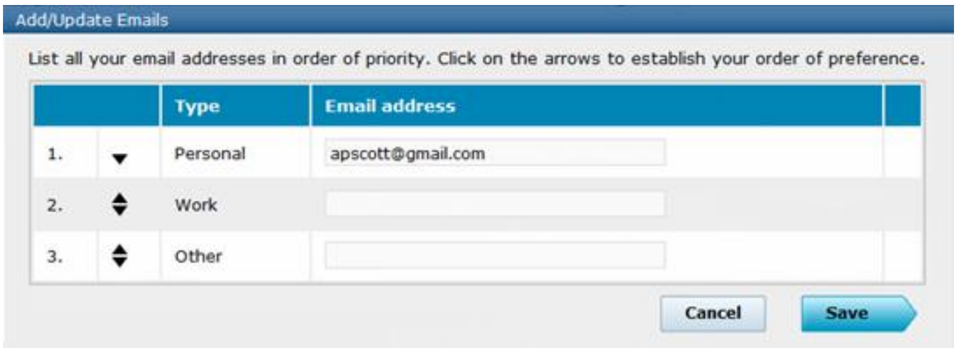


This is then displayed as a 'care of' (or c/o);

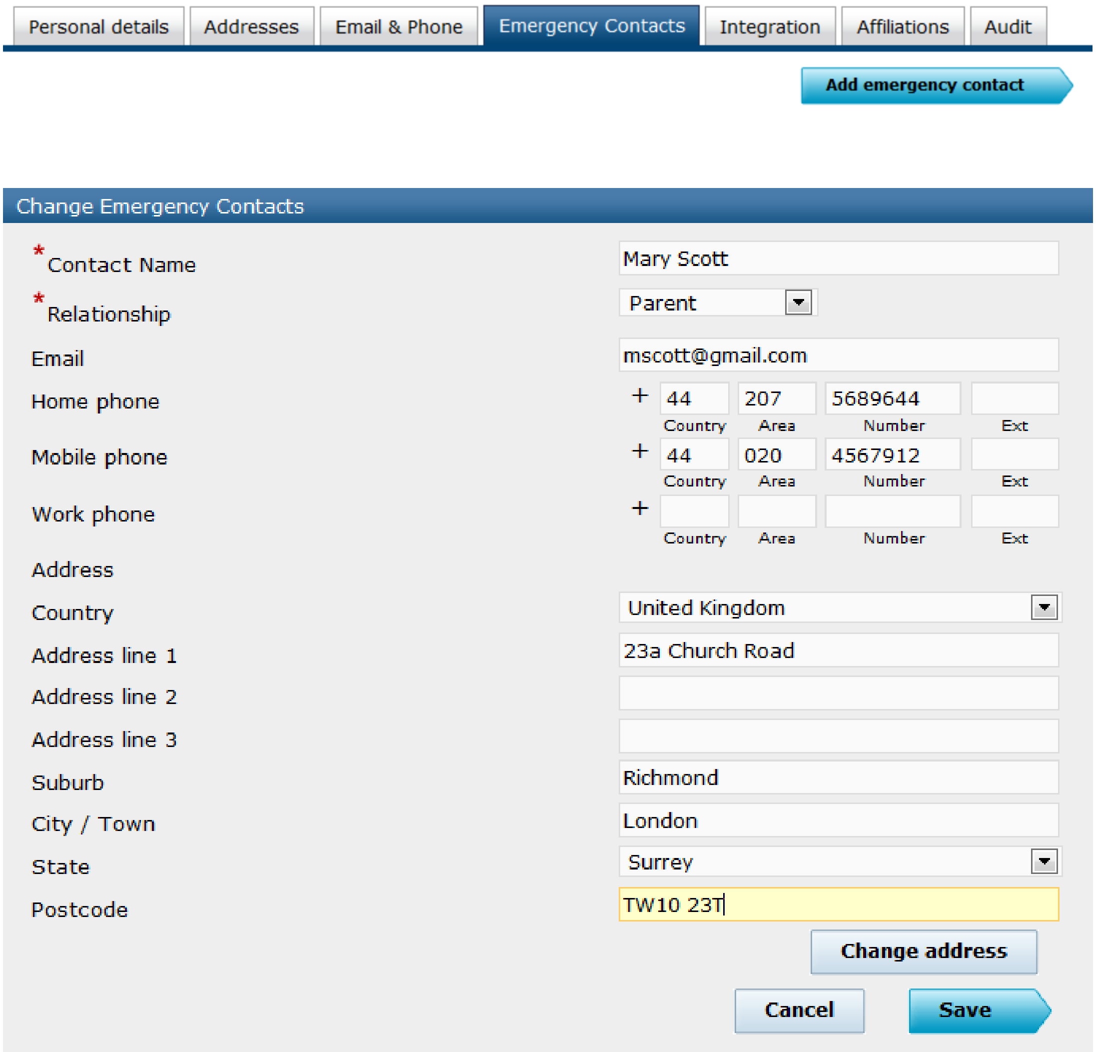
Add Update Email & Phone – IMS







Add / Update Emergency Contact Details – IMS

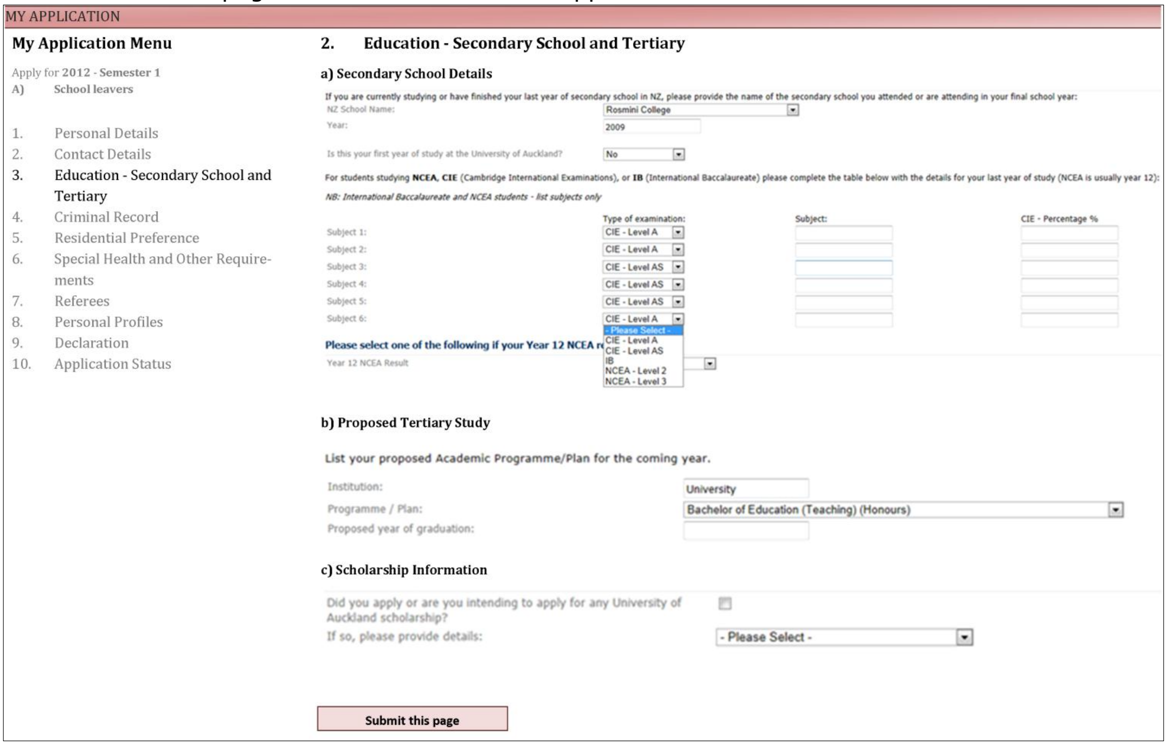


Other Application Information

1. NCEA Test Results and School Details

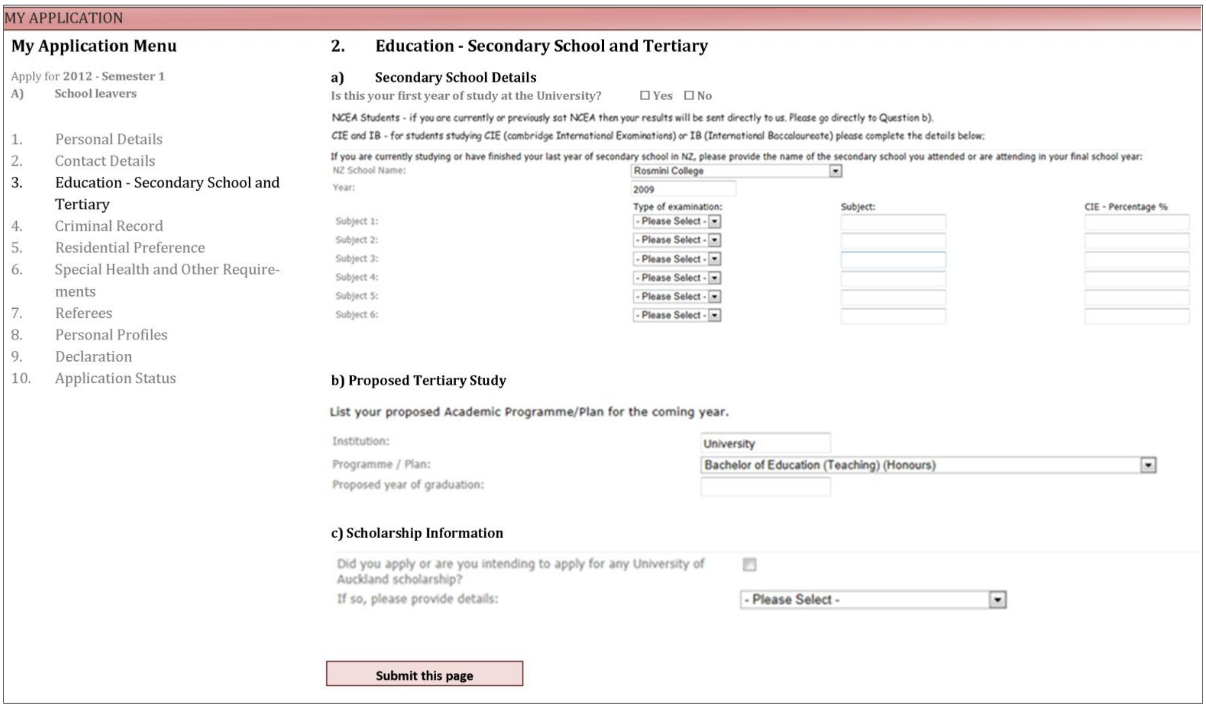
Section 3 of the UAM Application (Section 4 for international applicants) requires Secondary School details and results, Proposed Tertiary Study and Scholarship information – some of which is asked for and stored in other University systems;

This is the current page in the Accommodation Application



For NZ School Leavers who have NCEA results – they will no longer be required to fill out section a) of this form – as we would have collected their NCEA result and school information from SAP at the time that entered their NSN.

However, the section will need to remain (and be re-worded accordingly). See sample below;



MY APPLICATION

My Application Menu

Apply for 2012 - Semester 1

A) School leavers

1. Personal Details
2. Contact Details
3. Education - Secondary School and Tertiary
4. Criminal Record
5. Residential Preference
6. Special Health and Other Requirements
7. Referees
8. Personal Profiles
9. Declaration
10. Application Status

2. Education - Secondary School and Tertiary

a) Secondary School Details

Is this your first year of study at the University? ☐ Yes ☐ No

NCEA Students - if you are currently or previously sat NCEA then your results will be sent directly to us. Please go directly to Question b).

CIE and IB - for students studying CIE (cambridge International Examinations) or IB (International Baccalaureate) please complete the details below:

If you are currently studying or have finished your last year of secondary school in NZ, please provide the name of the secondary school you attended or are attending in your final school year:

NZ School Name: Rosmini College

Year: 2009

	Type of examination:	Subject:	CIE - Percentage %
Subject 1:	- Please Select -		
Subject 2:	- Please Select -		
Subject 3:	- Please Select -		
Subject 4:	- Please Select -		
Subject 5:	- Please Select -		
Subject 6:	- Please Select -		

b) Proposed Tertiary Study

List your proposed Academic Programme/Plan for the coming year.

Institution: University

Programme / Plan: Bachelor of Education (Teaching) (Honours)

Proposed year of graduation:

c) Scholarship Information

Did you apply or are you intending to apply for any University of Auckland scholarship?

If so, please provide details: - Please Select -

Submit this page

2. Scholarship Information

Section c) of this application page should stay as it is. However, we will also be importing any SAP scholarship information (i.e. whether an applicant has applied for a scholarship, and if so, the name of the scholarship) for accommodation applicants from the new Scholarship Management system. These will only relate to centrally administered scholarships and not those managed by the faculties. The Scholarship Management online application system is currently in development and is unlikely to be available to provide this information at Go Live. Instead this integration will be turned on when Scholarship Management is implemented. Given that the above question will remain in the application form, UAM will continue to capture this information manually from applicants.

3.4.3 Synchronising Other Personal Data

IMS Changes and Affiliation data

When an affiliation (of interest to UAM) is added to or removed from a person that exists in the UAM database (i.e. a current applicant, and past or present University resident), a message should be published from the IMS and subscribed to by UAM to update the person's record. The IMS will send all Affiliation changes to UAM for all IMS identities, but will update only those persons that exist in the UAM database, via an API. Note: UAM is interested in any change to a person's identity record that is not captured by the processes in Section 3.4.1 and 3.4.2 of this document.

The Affiliations of interest to UAM are;

a. Applicant
b. Undergraduate Student
c. Postgraduate Student
d. Doctoral Student
e. Alumni

Photos

Along with the affiliation data - photos should also be published to UAM, and re-sent whenever they are updated in the ID card system – for students who are current accommodation applicants or residents (as with the affiliation messages). Currently, there is an outbound message from the ID card system which UAM will subscribe to receive newly added or updated photos.

Note: Given the size of the files in relation to photos it will be necessary for the UAM administrators to clean out photos that are not required on a regular basis, e.g. for those applicants that never become residents.

3.4.4 Reports

Once Affiliation data is interfaced into UAM, the business will be able to better identify persons who are either currently staying at a University residence and are not entitled to (as they are no longer or never were a student) or have an outstanding accommodation application but have been refused entry to a programme of study at the University. Reports are required to be written that will use the affiliation information of a person in the UAM to determine their eligibility to accommodation.

Data required for the Report will be;

a. UAM ID number
b. UniversityID
c. Legal Name
d. Resident Status
e. Resident Year
f. Enrolment Term
g. Enrolment Status
h. Residence
i. Current Affiliations

3.5 Data Requirements and Transformations

3.5.1 Personal Details

Field Description	Field Type	Req	UAM Record Name	UAM Field Name	IMS Record Name	IMS Field Name	Comments
Surname	Char 40	Y	Entry	NameLast	PERSONNAME	LASTNAME Where NAMETYPE = PRI	IMS Legal last name
First Name	Char 40	Y	Entry	NameFirst	PERSONNAME	FIRSTNAME Where NAMETYPE = PRI	
Middle Name	Char 40?	N	Entry	NameOther?	PERSONNAME	MIDDLENAME Where NAMETYPE = PRI	
Preferred Name	Char 40	Y	Entry	NamePreferred		FIRSTNAME LASTNAME Where NAMETYPE = PRF	IMS Preferred Names
University ID	Char 30	Y	Entry	ID1?	PERSONEXTERNAL IDENTIFIER	IDENTIFIERVALUE Where .DENTIFIERTYPE = 'UniversityID'	Will be required – and automatically populated
Gender	Int?	Y	Entry	GenderEnum (Edit Table)	PERSON	GENDER	F, M or U (Issue?)
Date of Birth	DateTime	Y	Entry	DOB	PERSON	DATEOFBIRTH	
Citizenship Status	Int?	Y	EntryDetail	Citizenship_ CountryID? (Edit Table)	PERSON	CITIZENSHIP	Country in IMS
Residency	Char?	Y			PERSON	RESIDENCY	
Ethnicity	Char 50	Y	EntryDetail	Ethnicity (Edit Table)	PERSONETHNICITY	PERSONETHNICITY	Can have multiple in IMS Mapping (Ethnic group Code)
Photo	Image		EntryDetail	PhotoImage	ID card System: PHOTO	CONTENT	Subscribe to ID card photo message – from ID card system (not IMS)

3.5.2 Address Details

Field Description	Field Type	Req	UAM Record Name	UAM Field Name	IMS Record Name	IMS Field Name	Comments
Mailing Address (AddressTypeID ='Mailing')							
Contact Name	Char 80	Y	Entry Address	Contact Name	PERSONPHYSICAL ADDRESS	CAREOF	Where ADDRESSTYPE = 'Mailing'
Street Address	Char 80	Y	Entry Address	Street (Note: Street2 also)	PERSONPHYSICAL ADDRESS	LINE1 & LINE2 & LINE3 & LINE4	
Suburb	Char 80		Entry Address	Street2?	PERSONPHYSICAL ADDRESS	SUBURB	
City/Town	Char 60	Y	Entry Address		PERSONPHYSICAL ADDRESS	CITY	
Region	Char 60	Y	Entry Address	StateProvince? (Edit Table)	PERSONPHYSICAL ADDRESS	COUNTY or STATE	
Country	Int	Y	Entry Address	Country_ID (Edit table)	PERSONPHYSICAL ADDRESS	COUNTRY	
Postal Code	Char 10	Y	Entry Address	ZipPostcode	PERSONPHYSICAL ADDRESS	POSTALCODE	
Phone and Email							
Telephone Number	Char 25	Y	Entry Address	Phone	PERSONPHONE	AREACODE|| PHONENUMBER	Where PHONETYPE = 'Semester'? Or 'Other'? or 'Home'
Preferred Flag?					PERSONPHONE	ORDERPREFERENCE	Where PHONETYPE = 'Semester'? Or 'Other'? or 'Home' and ORDERPREFERENCE = 1
Mobile Phone	Char 25		Entry Address	PhoneMobileCell	PERSONPHONE	AREACODE|| PHONENUMBER	Where PHONETYPE = 'Cellular'

Preferred Flag?					PERSONPHONE	ORDERPREFERENCE	Where PHONETYPE = 'Cellular' ORDERPREFERENCE = 1
Email	Char 100		Entry Address	Email	PERSONEMAIL ADDRESS	EMAIL	Where EMAILYPE = 'Home'? *** Allow it to be editable?

Home Address (AddressTypeID ='Home')

Contact Name	Char 80	Y	Entry Address	Contact Name	PERSONPHYSICAL ADDRESS	CAREOF	Where ADDRESSTYPE = 'Home'
Street Address	Char 80	Y	Entry Address	Street (Note: Street2 also)	PERSONPHYSICAL ADDRESS	LINE1 & LINE2 & LINE3 & LINE4	
Suburb	Char 80	Y	Entry Address	Street2?	PERSONPHYSICAL ADDRESS	SUBURB	
City/Town	Char 60	Y	Entry Address		PERSONPHYSICAL ADDRESS	CITY	
Region	Char 60	Y	Entry Address	StateProvince? (Edit Table)	PERSONPHYSICAL ADDRESS	COUNTY or STATE	
Country	Int	N	Entry Address	Country_ID (Edit table)	PERSONPHYSICAL ADDRESS	COUNTRY	
Postal Code	Char 10	Y	Entry Address	ZipPostcode	PERSONPHYSICAL ADDRESS	POSTALCODE	

3.5.3 Emergency Contact Address

Contact Name	Char 80	Y	Entry Address	Contact Name	PERSONEMERGENCY CONTACT	CONTACTNAME	
Emergency Contact Relationship	Char 50	Y	Entry Address	Relationship	PERSONEMERGENCY CONTACT	RELATIONSHIP	Mapping required? See IMS Values below
Street Address	Char 80	Y	Entry Address	Street	PERSONEMERGENCY CONTACT	CONTACTCAREOF|| CONTACTBUILDING CONTACTSTREET	Note: each field has 50 characters!

Suburb	Char 80	N	Entry Address	Street2?	PERSONEMERGENCY CONTACT	CONTACTSUBURB	
City/Town	Char 60	Y	Entry Address		PERSONEMERGENCY CONTACT	CONTACTCITY	
Region	Char 60	Y	Entry Address	StateProvince? (Edit Table)	PERSONEMERGENCY CONTACT	CONTACTCOUNTY or CONTACTSTATE	
Country	Int	Y	Entry Address	Country_ID (Edit Table)	PERSONEMERGENCY CONTACT	CONTACTCOUNTRY	
Postal Code	Char 10	Y	Entry Address	ZipPostcode	PERSONEMERGENCY CONTACT	POSTALCODE	
Telephone Number	Char 25	Y	Entry Address	Phone	PERSONEMERGENCY CONTACT	PHONEAREACODE|| PHONENUMBER	Where PHONETYPE = ??? (Home or Work)
Mobile Phone	Char	N	Entry Address	PhoneMobileCell	PERSONEMERGENCY CONTACT	PHONEAREACODE|| PHONENUMBER	Where PHONETYPE = 'Mobile'
Email	Char	N	Entry Address	Email	PERSONEMERGENCY CONTACT	EMAIL???	

3.5.4 School Details and NCEA Results

Field Description	Field Type	Req	UAM Record Name	UAM Field Name	SAP Record Name	SAP Field Name	Comments
NSN		Y			SAD_NCEA_S_NZL	SCC_NSN	
School Name					EXT_ORG_TBL	DESCR	
Last Year of School					SAD_NCEA_S_NZL	SAD_UEBS_YEAR	
NCEA level					SAD_NCEA_STDNZL	SAD_NCEA_LEVEL	
Subject					SAD_NCEA_STDNZL	TEST_COMPONENT	

Standard?					SAD_NCEA_STDNZL	SAD_NCEA_STD_CODE	
Result?					SAD_NCEA_STDNZL	SAD_NCEA_RESULT	

3.5.5 Affiliations

Field Description	Field Type	Req	UAM Record Name	UAM Field Name	LDAP Record Name	LDAP Field Name	Comments
Student ID					?	EMPLID	
Affiliation?					?	GROUP?	Affiliation values such as APPLICANT, ALUMNI, DOCTORATE, POSTRAG, UNDERGARD

3.5.6 Scholarship Information

Field Description	Field Type	Req	UAM Record Name	UAM Field Name	SAP Record Name	SAP Field Name	Comments
Student ID					RSH_AWDSTAT_ANZ	EMPLID	
Year					RSH_AWDSTAT_ANZ	RSH_OFFER_YEAR	
Status					RSH_AWDSTAT_ANZ	RSH_SCHOLAR_STATUS	e.g. Applied, Active, Offer etc
Description					RSH_SCH_DTL_ANZ	RSH_DESCRFORMAL	Description of Scholarship

3.5.7 Reports

Field Description	Field Type	UAM Record Name	UAM Field Name	Comments
Entry ID	Int	Entry	EntryID	7 character code (sent from IMS)
Legal Name	Char 40	Entry	NameFirst|| NameOther ||NameLast	Concatenate Name fields (40 characters each)
Resident Status	Char 50	EntryDetail	ResidentStatus	
Resident Year	Char 50	EntryDetail	ResidentYear	
Enrolment Term	Char 50	EntryDetail	EnrollmentTerm	
Enrolment Status	Char 50	EntryDetail	EnrollmentStatus	
Residence		???		Current Residence
Affiliations	Char 12			

3.5.8 Configuration Data

3.5.8.1 IMS Relationship Values:

Currently the Accommodation application has a free text field for the description of an applicant's relationship to their Emergency Contact. The IMS Emergency Contact is validated against the following 'relationship' values;

Relationship Descriptions

1. Aunt
2. Brother
3. Daughter
4. Employee
5. ExSpouse
6. Father
7. Father-in-Law
8. Flatmate
9. Friend
10. Grandchild
11. Grandfather
12. Grandmother
13. Guardian
14. Mother
15. Mother-in-Law
16. Neighbour
17. Nephew
18. Niece
19. Non-Qualified Adult
20. Other
21. Other Relative
22. Partner
23. Self
24. Sister
25. Son
26. Spouse
27. Uncle

3.5.8.2 Ethnicity

Below is a comparison of the Ethnicity values in the IMS and the current UAM application. UAM will be required to bring their values in line with the IMS.

IMS Value	UAM Value	Comments
Australian	Australian	
British and Irish	British	UAM has these listed separately
Cambodian	Cambodian	
Chinese	Chinese	
Cook Island Maori	Cook Island Maori	
Dutch	Dutch	
Fijian	Fijian	

Filipino	Filipino	
German	German	
Greek	Greek	
Indian	Indian	
	Irish	See above
Italian	Italian	
Japanese	Japanese	
Korean	Korean	
Latin American/Hispanic	Latin American/Hispanic	
	Malaysian	Not in IMS
Middle Eastern	Middle Eastern	
Niuean	Niuean	
North American	North American	
NZ European/Pakeha	NZ European/Pakeha	
NZ Maori	NZ Maori	
No Response		Not in UAM
Other		Not in UAM
Other African	Other African	
Other Asian	Other Asian	
Other European	Other European	
Other Pacific Island	Other Pacific Island	
Other South East Asian	Other South East Asian	
Polish	Polish	
African	South African/African	No South African group in IMS
	South African/European	No South African group in IMS
Samoan		No Samoan group in UAM
South Slav	South Slav	
Sri Lankan	Sri Lankan	
Tokelauan	Tokelauan	
Tongan	Tongan	
Vietnamese	Vietnamese	

3.5.8.3 Citizenship & Residency

In UAM there are currently only 4 Citizenship Groups;

1. NZ Citizen
2. Australia Citizen
3. Permanent Resident
4. Overseas

In the IMS Citizenship equates to the Country Code on an individual's passport – currently there are 255 country codes (per SAP Country table). The residency will be derived from the Citizenship (in the IMS), i.e. New Zealand and Australian citizens will be given residency of NZ and Australia. All other persons will be required to disclose whether or not they are permanent residents or if not, they will be deemed to be 'Overseas'.

3.5.8.4 Address Regions

In UAM there are currently only 16 Regions available against the physical address. These are;

1. Auckland
2. Bay of Plenty
3. Canterbury
4. Gisborne
5. Hawkes Bay
6. International
7. Manawatu Wanganui
8. Marlborough
9. Nelson
10. Northland
11. Otago
12. Southland
13. Taranaki
14. Waikato
15. Wellington
16. West Coast

In the IMS there is no Region validation. Instead, for NZ address there is in-built validation as you enter the address. For some foreign address there is State or County validation based on the country selected.

3.6 Configuration Requirements

Given the difference in some of the values stored in the IMS for Countries, Citizenship and Ethnicity, it will be necessary to change the current edit table values in UAM to be in alignment. Additionally, if the affiliation is to be brought into UAM there may be some configuration required to store these.

3.7 Non-Functional Requirements

3.7.1 Security

With the change to have users sign into the UAM Application using their IMS credentials, the sign-on security within UAM will have to be re-written to accept the IMS ID and passwords. It will be no less secure than it is currently and there should be no need for any change to roles and security profiles within UAM. This will be detailed as part of technical specification.

3.7.2 Performance

A change to master login information and personal and contact data within the IMS adds extra complexity and with it, potential for performance degradation. Any web service or messaging from the IMS to UAM should happen in near-real time, as it does with similar interfaces between the IMS and the AfUE or IMS and SAP. The users should not notice that they are in fact in another system and any transferring between the two should be seamless.

3.7.3 Training

Training will be required for UAM staff in regards to how the IMS should be used by Accommodation Applicants. There may be additional affiliation functionality within UAM that staff need to be trained in also.

3.7.4 User Documentation

This specification will provide the necessary information for staff to understand any new functionality and how the Application process will work once integrated with the IMS.

3.7.5 On-going support and maintenance

The UAM system is currently and will continue to be supported by SMS. The Integration with the IMS will be supported by IT Service.

3.7.6 Technical Approach – API's and Web Services

The UAM system has a number of Application Programming Interfaces (API) available to load data from external systems. UAM will subscribe to the existing IMS Person message2 using the University's Enterprise Service Bus (ESB) to create a web service to pass and filter the data coming from the IMS, which will then be processed by the relevant API.

3.8 Testing

3.8.1 Test Scenarios

Req #	Test Scenario	Expected Outcome
1	Add a new Accommodation Application for a new user – not previously registered with the University	Accommodation Application submitted – UAM and IMS are linked using the University ID
2	Modify an Accommodation Application for a new Applicant using IMS credentials to log in	Can access Accommodation Application using IMS login credentials
2	Add an Accommodation Application for a user who has previously filed an application in the AfUE – and does not have all the necessary personal and contact details in the IMS	Can create an Accommodation Application and required fields show up on the personal and contact details pages in the UAM application.
2	Add an Accommodation Application for a user who was a Friend of the university and already has an University ID and password	Can create an Accommodation Application and required fields show up on the personal and contact details pages in the UAM application.
2	Change some personal details for an Applicant while completing step1 of the UAM application	Personal detail changes can be made from using the link to the IMS. Upon save the user is returned to the UAM Accommodation Application where the changes are reflected on the personal Details Page.
2	Change some contact details for an Applicant while completing step2 of the UAM application	Contact detail changes can be made from using the link to the IMS. Upon save the user is returned to the UAM Accommodation Application where the changes are reflected on the Contact Details Page.
3	Change some personal details for an Applicant directly in the IMS	Personal detail changes are sent to UAM and update the database.
3	Change some contact details for an Applicant directly in the IMS	Contact detail changes are sent to UAM and update the database.
4	Change an applicant's affiliations from applicant to student	Personal detail changes are sent to UAM and update the database.

4,5	Remove an accommodation applicant's 'applicant' affiliation in SAP, i.e. change their Programme Status to ensure it is NOT one of the following; AD(Approved) AP(Pending) PM(Prematriculant) WT(Waitlisted) Then run the Affiliation Report	Personal detail changes are sent to UAM and update the database. The report runs successfully and shows the applicant whose status has changed.
4,6	Drop a student (who is currently in a University Residence) from their programme of study. Then run the Affiliation Report	Personal detail changes are sent to UAM and update the database. The report runs successfully and shows the applicant who has been dropped.
11	Create an application for a NCEA student	The Education page of the application will not ask for secondary school details
11	Create an application for a CIE student	The secondary school details and results questions on the Education page of the application will be displayed
10	Upload a photo for an accommodation applicant, in the ID card system	A photo will be sent to the UAM database for an existing applicant
12	Enter a scholarship application in SAP for a person who has an active accommodation application.	Scholarship information will be passed from SAP into the UAM database.

4 Approval and Change Control

Version #	**Description of Change**	**Author**
1.0	Initial draft	
2.0	Update following meetings with Accommodation staff	
3.0	Update following meeting with Development and Solutions Architects staff	
4.0	Update following conversations with the Integration Architecture office	

Review	
Date	**Name and Position**

Approval		
Date	**Name and Position**	**Signed**

G
UAM/IMS Integration Use Case Model

Experiment Code: UCM2012

UAM / IMS Integration

Use Case Model

Contents

1. Summary

1.1 Background

The University Accommodation Management system, UAM, is a specialised student housing solution. Applicants for student accommodation register by completing an online application via a link from the University website.

The Identity Management System (IMS) was introduced to the University in 2008 and is the University's system of record for identity information and primary web authentication. There is currently no automated integration between UAM and the IMS.

In order to bring some efficiency and provide a better user experience for students it is desirable to integrate the UAM system with the IMS – ensuring that a student only needs to register and update personal data in one system. Additionally, it would be beneficial to pass various affiliation information from other University systems to UAM to keep them informed of any student's status changes.

1.2 Glossary of terms

Term	**Meaning**
UAM	University Accommodation Management (UAM) system is the University's specialised student housing solution.
SAP	The Student Administration Platform.
IMS	The Identity Management System (IMS) is the University's master repository for Personal Data. It stores and maintains details of all persons that the University has a relationship with, including students, staff, visitors, alumni and contractors.
AfUE	Application for University Entrance. This is an application for recording applications for admission to programmes of study at the University. It interfaces with the IMS for the Personal and Contact details of the applicants.
ULN	University Login Name – assigned to persons in the IMS when an Identity becomes resolved.

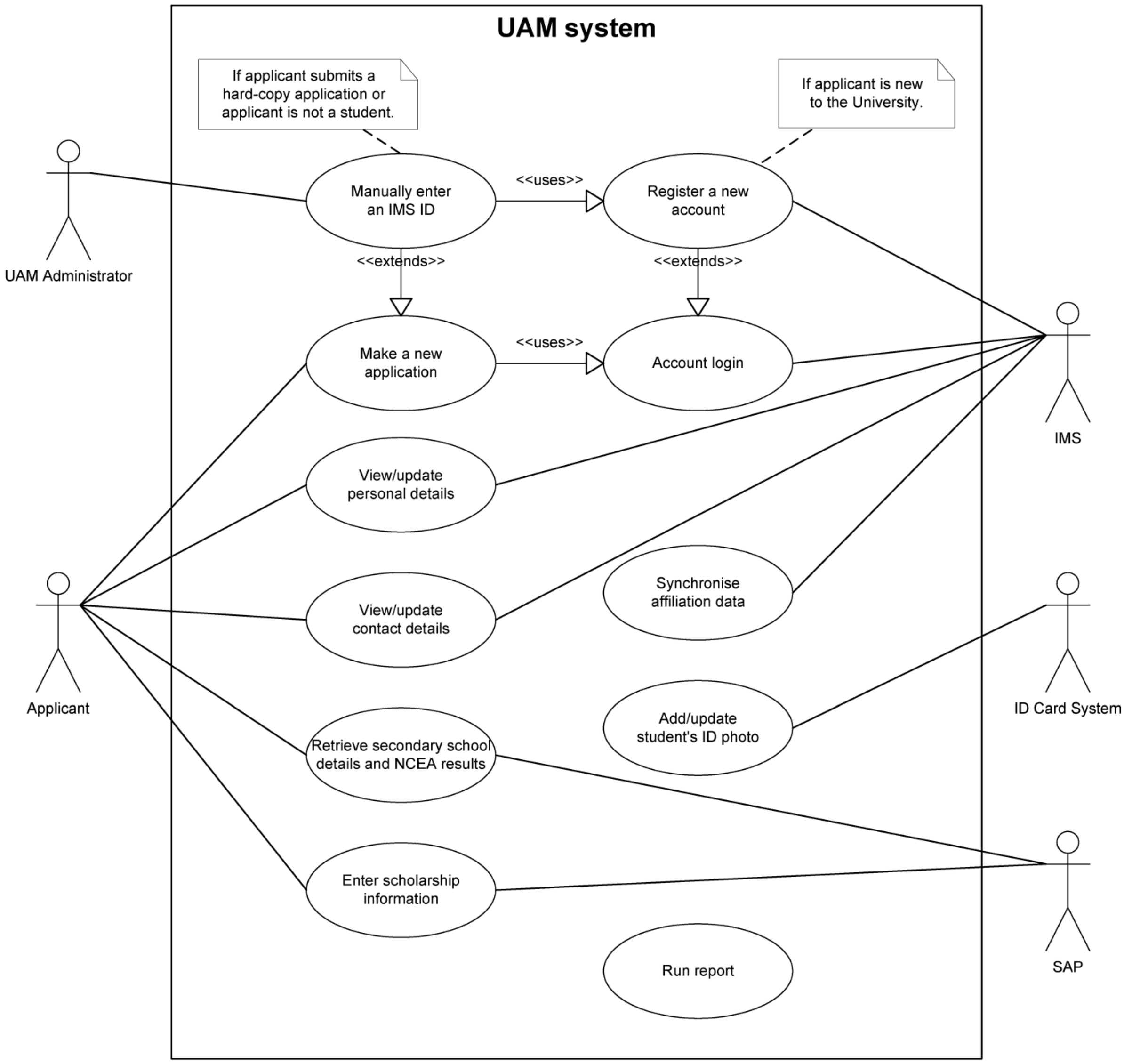
2. Solution Overview

When a prospective or current student wants to submit an online Application for Accommodation at the University they will use the links currently provided on various web pages, e.g. the University Home page 'Accommodation' link, or via the Application for University Entrance (Accommodation Services link).

The solution for accommodation follows the pattern already established for similar web forms like the Application for University Entrance. Applicants visiting the Application for Accommodation for the first time will be required to register themselves in the IMS and provide all the necessary personal data. The applicant will then be transferred back to the accommodation form where their personal data will be displayed 'read only'. A link back to the IMS will be provided in order that applicants can update personal data at any time.

Additionally, it would be useful for Accommodation staff to have basic student-related information provide to UAM from other systems, to assist them in processing an applicant's request for accommodation.

3. Use Case Diagram



4. Actors

4.1 Applicant

Actor	Applicant
Description	The person who wants to submit an application for Accommodation at the University.
Example	Prospective or current student

4.2 UAM administrator

Actor	UAM administrator
Description	The person who processes an applicant's request for accommodation.
Example	Accommodation staff

4.3 IMS

Actor	IMS
Description	The Identity Management System (IMS) is the University’s master repository for Personal Data. It stores and maintains details of all persons that the University has a relationship with, including students, staff, visitors, alumni and contractors.

4.4 ID card system

Actor	ID card system
Description	The University's ID card system.

4.5 SAP

Actor	SAP
Description	The Student Administration Platform.

5. Use Cases

5.1 Make a new application

Actors	Applicant or UAM administrator
Trigger	The applicant wants to submit an application for Accommodation at the University.
Prerequisites	None.
Post-conditions	The applicant or the UAM administrator will have completed an Accommodation Application successfully.
Normal flow of events	1. The 'Account Login' section on the University Accommodation Home page directs the user to the IMS for Sign in (see use case "Account login"). 2. Once the applicant has either logged in or registered (in the IMS) and completed their verification, the applicant can begin a new Application, by selecting the 'My Application' link from the home page. 3. The applicant selects their application type and clicks on 'Save and Continue' – they will be transferred to the 'Personal Details' page (see use case "View/update personal details"). 4. The applicant is then transferred to the 'Contact Details' page in the Accommodation Application (see use case "View/update contact details") after viewing or updating their personal details. 5. The applicant completes the remaining parts of the Accommodation application.
Variations	If the applicant submits a hard-copy application or the applicant is not a student, a UAM administrator will enter the application for them (see use case "Manually enter IMS ID").
Use Case associations	The related use cases are: • Account login • View/update personal details • View/update contact details • Manually enter an IMS ID

5.2 Account login

Actors	Applicant, IMS
Trigger	The 'Account Login' section on the University Accommodation Home page directs the user to the IMS for Sign in.
Prerequisites	None.
Post-conditions	The applicant will have been successfully logged in.
Normal flow of events	1. The 'Account Login' section on the University Accommodation Home page directs the user to the IMS for Sign in. 2. For someone who has previously registered with the university, they can use their University ID (7 character number), ULN (if they are already a student) or the personal email address they used to create their account. 3. The applicant is being redirected to the 'Welcome' page in the Accommodation Application once signed in.
Variations	If the applicant is new to the University then the applicant will be taken to the IMS registration screen (see use case “Register a new account”).
Use Case associations	This use case is extended by the “Register a new account” use case.

5.3 Register a new account

Actors	Applicant or UAM administrator, IMS
Trigger	1. The applicant is being taken to the IMS registration screen before completing an Accommodation Application, or 2. The UAM administrator creates an IMS identity on behalf of an applicant.
Prerequisites	The applicant is new to the University.
Post-conditions	The applicant or the UAM administrator will have successfully completed the registration. The applicant can return to the Accommodation portal – Home page and lodge an application.
Normal flow of events	1. The applicant is taken to the IMS registration screen. 2. The applicant is required to enter the following; • Email Address • First Name • Last Name • Password 3. Following completion of this form and acceptance of the terms and conditions the applicant will be sent a confirmation email. 4. The applicant is required to complete the verification by clicking on the 'Complete Your Registration' link and populating the following information (some of which is compulsory). • Title • Middle Names • Preferred Name • Previous Name • Mobile Phone Number (required) • Home Phone Number (required) • Correspondence Address (required) • Gender (required) • Date of Birth (required) • Citizenship (required) • Residency (if not an NZ citizen) • Ethnicity (required) • Emergency Contact details (required) • National Student Number (NSN) 5. Once this page is complete, the applicant can return to the

	Accommodation portal – Home page and lodge an application.
Variations	None.
Use Case associations	None.

5.4 Manually enter an IMS ID

Actors	UAM administrator
Trigger	1. The UAM administrator enters an application on behalf of an applicant, or 2. The UAM administrator processes applications for non-student accommodation residents, who are currently entered into UAM through the Administration pages, or 3. The UAM administrator processes applications for those students that submit a hard-copy application.
Prerequisites	None.
Post-conditions	The UAM administrator will have successfully gathered the personal data from the IMS.
Normal flow of events	1. The UAM administrator enters an application on behalf of an applicant. 2. The UAM administrator creates the identity in the IMS first (see use case "Register a new account"). 3. The UAM administrator enters the IMS ID number into UAM manually. 4. The UAM administrator pushes the person message from the IMS in order to populate the required personal data fields.
Variations	None.
Use Case associations	This use case includes the use case "Register a new account".

5.5 View/update personal details

Actors	Applicant, IMS
Trigger	The applicant is being transferred to the 'Personal Details' page in the Accommodation Application.
Prerequisites	The applicant has logged in.
Post-conditions	The information on the 'Personal Details' page in the Accommodation Application will be updated with whatever was saved in the IMS. This will be a near real-time update.
Normal flow of events	1. The applicant is being transferred to the 'Personal Details' page. All fields are display only on this page. 2. The applicant can maintain their own data by clicking the link 'Update Personal Details' on the page. This link will transfer the user to the IMS (in a new browser window). 3. The applicant returns to the 'Personal Details' page after updating their information in the IMS.
Variations	None.
Use Case associations	None.

5.6 View/update contact details

Actors	Applicant, IMS
Trigger	The applicant is being transferred to the 'Contact Details' page in the Accommodation Application.
Prerequisites	The applicant has logged in.
Post-conditions	The information on the 'Contact Details' page will be updated with what was saved in the IMS.
Normal flow of events	1. The applicant is being transferred to the 'Contact Details' page in the Accommodation Application - all fields are read only with three separate buttons to link to different parts of the IMS for adding and/or update Addresses, Phone & Email and Emergency Contacts details. 2. A new browser window will be opened in the IMS on the appropriate page – and when saved will push a near-real time message to the Accommodation system to update anything that has changed. 3. The applicant returns to the 'Contact Details' page after updating their contact information in the IMS.
Variations	None.
Use Case associations	None.

5.7 Synchronise affiliation data

Actors	IMS
Trigger	When affiliations are updated (or when a user is initially sent to the Accommodation system).
Prerequisites	None.
Post-conditions	The UAM system has up to date information relating to an applicant's (or current resident's) status with the University.
Normal flow of events	1. When an affiliation (of interest to UAM) is added to or removed from a person that exists in the UAM database (i.e. a current applicant, and past or present University resident), a message is published from the IMS and subscribed to by UAM to update the person's record. 2. The IMS sends all Affiliation changes to UAM for all IMS identities, but updates only those persons that exist in the UAM database, via an Application Programming Interface (API). The Affiliations of interest to UAM are; • Applicant • Undergraduate Student • Postgraduate Student • Doctoral Student • Alumni
Variations	None.
Use Case associations	None.

5.8 Add/update student's ID photo

Actors	ID card system
Trigger	When a new photo is added or updated for a student who is a current accommodation applicant or resident.
Prerequisites	None.
Post-conditions	The photos for students are being updated in the UAM system.
Normal flow of events	1. A message is published from the university ID card system every time a new photo is added for a student. 2. The UAM system updates the photos for students who are current accommodation applicants or residents whenever it receives a message from the ID card system. This updates the UAM database directly via an API.
Variations	None.
Use Case associations	None.

5.9 Retrieve secondary school details and NCEA results

Actors	Applicant, SAP
Trigger	When an applicant selects the 'School Leaver' option for the Accommodation Application type.
Prerequisites	The applicant has logged in.
Post-conditions	Secondary School details and NCEA results (from Year 12 onwards) will be sourced from SAP data (and populated in the UAM database) if this is available at the time an applicant completes the form – and we have received their National Student Number (NSN). Any stored NCEA exam results in SAP will not be displayed to the applicant.
Normal flow of events	1. When the applicant selects the 'School Leaver' option on the UAM application form, the applicant will be prompted for their National Student Number (NSN) if it is not already stored against their IMS identity. 2. When the NSN number is entered, or on confirmation that it has been entered, a web service will pull all relevant NCEA test data and school information from the SAP database. The applicant is no longer required to fill out the section on secondary school details in the Accommodation Application.
Variations	None.
Use Case associations	None.

5.10 Enter scholarship information

Actors	Applicant, SAP
Trigger	When an applicant answers the section on University scholarship as part of the Education question in the Accommodation Application.
Prerequisites	The application has logged in.
Post-conditions	Scholarship information will be pre-populated in the UAM database, from SAP, if it exists at the time an applicant completes the form.
Normal flow of events	1. The applicant answers information on whether he or she has previously applied for or intends to apply for a scholarship in the Accommodation Application. 2. If an applicant has previously applied the information relating to this application is currently stored in SAP (though in the future this will be in Scholarship Management system) and could be retrieved by a web service and posted directly to the UAM database via an API. 3. The current scholarship question will be left in space, for cases where applicants intend to apply or have applied for a scholarship that is not centrally managed, i.e. some faculties manage their own scholarships.
Variations	None.
Use Case associations	None.

5.11 Run report

Actors	
Trigger	Once Affiliation data is interfaced into UAM, the business will be able to better identify persons who are either currently staying at a University residence and are not entitled to (as they are no longer or never were a student) or have an outstanding accommodation application but have been refused entry to a programme of study at the University. Reports are required to be written that use the affiliation information of a person in the UAM to determine applicant's eligibility to accommodation. The requirements for the report is: • To identify persons with current accommodation applications who are not current students and whose applications for admission have been declined. • To identify current University residents who are no longer active students (or were never enrolled).
Prerequisites	None.
Post-conditions	Data required for the Report will be; • UAM ID number • UniversityID • Legal Name • Resident Status • Resident Year • Enrolment Term • Enrolment Status • Residence • Current Affiliations
Normal flow of events	
Variations	None.
Use Case associations	None.

References

[1] A. Abran and P. Bourque. *SWEBOK: Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society, 2004.

[2] A. Al-Rawas and S. Easterbrook. Communication problems in requirements engineering: A field study. In *Proc. of Conf. on Prof. on Awareness in Software Engineering*, pages 47–60, 1996.

[3] M.J. Albers. Signal to noise ratio of information in documentation. In *Proceedings of the 22nd Annual International Conference on Design of Communication: The Engineering of Quality Documentation*, pages 41–44. ACM, 2004.

[4] M. Allaby, editor. *A Dictionary of Earth Sciences*. Oxford University Press, 2008.

[5] B. Anda, D. Sjøberg, and M. Jørgensen. Quality and understandability of use case models. In J.L. Knudsen, editor, *ECOOP 2001 Object-Oriented Programming*, volume 2072 of *Lecture Notes in Computer Science*, pages 402–428. Springer Berlin Heidelberg, 2001.

[6] J. Aranda, N. Ernst, J. Horkoff, and S. Easterbrook. A framework for empirical evaluation of model comprehensibility. In *Proceedings of the International Workshop on Modeling in Software Engineering*, page 7. IEEE Computer Society, 2007.

[7] A. Aurum and C. Wohlin. Requirements engineering: Setting the context. In *Engineering and Managing Software Requirements*, pages 1–15, 2005.

[8] N.F. Awad and M.S. Krishnan. The personalization privacy paradox: An empirical evaluation of information transparency and the willingness to be profiled online for personalization. *MIS Quarterly*, 30(1):pp. 13–28, 2006.

[9] Basle Committee on Banking Supervision. Enhancing bank transparency. `http://www.bis.org/publ/bcbs41.pdf`, March 2013.

[10] K. Bickerstaff, R. Tolley, and G. Walker. Transport planning and participation: The rhetoric and realities of public involvement. *Journal of Transport Geography*, 10(1):61 – 73, 2002.

[11] C. Bird. Top 10 tips for better agile. *Information Professional*, 2(6):33 –36, 2005.

[12] P.A. Boghossian. The transparency of mental content. *Philosophical Perspectives*, 8:33–50, 1994.

[13] T.D. Breaux, A.I. Anton, K. Boucher, and M. Dorfman. Legal requirements, compliance and practice: An industry case study in accessibility. In *16th IEEE International Requirements Engineering. RE '08.*, pages 43–52, 2008.

[14] R. Brooks. Towards a theory of the comprehension of computer programs. *International Journal of Man-Machine Studies*, 18(6):543–554, 1983.

[15] P. Cane and J. Conaghan, editors. *The New Oxford Companion to Law.* Oxford University Press Inc, 2008.

[16] D.O. Case. *Looking for information: A survey of research on information seeking, needs, and behavior.* Emerald Group Pub Limited, 2012.

[17] S. Cerri. Effective communication skills for engineers. In *Proceedings of the 2000 IEEE Engineering Management Society*, pages 625–629, 2000.

[18] R.N. Charette. Why software fails. *IEEE Spectrum*, 42(9):42 – 49, September 2005.

[19] A.H. Chayes and A. Chayes. Regimes architecture: Elements and principles. In J.E. Nolan, editor, *Global Engagement : Cooperation and Security in the 21st Century*, page 81. Washington, D.C. : Brookings Institution, 1994.

[20] R. Clarke. Internet privacy concerns confirm the case for intervention. *Communications of the ACM*, 42(2):60–67, February 1999.

[21] R. Cohen and J.S. Hiller. What's mine is mine; what's yours is mine: Private ownership of ICTs as a threat to transparency. *Ethics and Information Technology*, 11(2):123–131, 2009.

[22] N. Condori-Fernández, M. Daneva, K. Sikkel, and An. Herrmann. Practical relevance of experiments in comprehensibility of requirements specifications. In *2011 First International Workshop on Empirical Requirements Engineering (EmpiRE)*, pages 21–28. IEEE, 2011.

[23] M.F. Costabile, D. Fogli, G. Fresta, P. Mussio, and A. Piccinno. Computer environments for improving end-user accessibility. In *Universal Access Theoretical Perspectives, Practice, and Experience*, pages 129–140. Springer, 2003.

[24] J. Coughlan and R.D. Macredie. Effective communication in requirements elicitation: A comparison of methodologies. *Requirements Engineering*, 7(2):47–60, 2002.

[25] H. Cramer, V. Evers, S. Ramlal, M. van Someren, L. Rutledge, N. Stash, L. Aroyo, and B. Wielinga. The effects of transparency on trust in and acceptance of a content-based art recommender. *User Modeling and User-Adapted Interaction*, 18:455–496, 2008.

[26] M.J. Culnan. The dimensions of accessibility to online information: Implications for implementing office information systems. *ACM Transactions on Information Systems (TOIS)*, 2(2):141–150, 1984.

[27] B. Curtis, H. Krasner, and N. Iscoe. A field study of the software design process for large systems. *Communications of the ACM*, 31(11):1268–1287, November 1988.

[28] L.M. Cysneiros and V.M.B. Werneck. An initial analysis on how software transparency and trust influence each other. In *Anais do WER09 - Workshop em Engenharia de Requisitos, Valparaso, Chile, Julho 16-17*, pages 27–32, 2009.

[29] F.S.C. Da Silva and J. Agusti-Cullell. Information flow. In *Information flow and knowledge sharing*, volume 2 of *Capturing Intelligence*, pages 65–88. Elsevier Science, 2008.

[30] L. Dabbish, C. Stuart, J. Tsay, and J. Herbsleb. Leveraging transparency. *IEEE Software*, 30(1):37–43, 2013.

[31] J. Daintith and E. Wright, editors. *A Dictionary of Computing*. Oxford University Press, 2008.

[32] T. Dale. How to plan a computer disaster. `http://www.cosc.canterbury.ac.nz/tony.dale/papers/incis.html`, December 2009.

[33] R. Dawson, P. Bones, B.J. Oates, P. Brereton, M. Azuma, and M.L. Jackson. Empirical methodologies in software engineering. In *STEP '03: Proceedings of the Eleventh Annual International Workshop on Software Technology and Engineering Practice*, pages 52–58, Washington, DC, USA, 2003. IEEE Computer Society.

[34] W. Dubbink, J. Graafland, and L. Liedekerke. CSR, transparency and the role of intermediate organisations. *Journal of Business Ethics*, 82:391–406, 2008.

[35] S. Dunlop, editor. *A Dictionary of Weather*. Oxford University Press, 2008.

[36] T. Dyba, B.A. Kitchenham, and M. Jorgensen. Evidence-based software engineering for practitioners. *IEEE Software*, 22(1):58–65, 2005.

[37] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian. Selecting empirical methods for software engineering research. In *Guide to Advanced Empirical Software Engineering*, pages 285–311. Springer London, 2008.

[38] J. Elia. Transparency rights, technology, and trust. *Ethics and Information Technology*, 11(2):145–153, 2009.

[39] A.F. Farhoomand and D.H. Drury. Managerial information overload. *Communications of the ACM*, 45(10):127–131, October 2002.

[40] K. Farooqui, L. Logrippo, and J. de Meer. The ISO reference model for open distributed processing: An introduction. *Computer Networks and ISDN Systems*, 27(8):1215–1229, 1995.

[41] G. Fitzgerald and N.L. Russo. The turnaround of the London ambulance service computer-aided despatch system (LASCAD). *European Journal of Information Systems*, 14(3):244–257, 2005.

[42] K.R. Fleischmann and W.A. Wallace. A covenant with transparency: Opening the black box of models. *Communications of the ACM*, 48(5):93–97, May 2005.

[43] K.R. Fleischmann and W.A. Wallace. Ensuring transparency in computational modeling. *Communications of the ACM*, 52(3):131–134, March 2009.

[44] S. Flowers. *Software failure: Management failure: Amazing stories and cautionary tales*. John Wiley & Sons, Inc., New York, NY, USA, 1996.

[45] M. Fowler. *UML distilled: A brief guide to the standard object modeling language*. Addison-Wesley Professional, 2004.

[46] R. Gajanayake, R. Iannella, and T. Sahama. Privacy by information accountability for e-health systems. In *6th IEEE International Conference on Industrial and Information Systems (ICIIS)*, pages 49 –53, August 2011.

[47] B. Gertler. Self-knowledge. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Winter 2008 edition, 2008.

[48] C. Ghezzi, M. Jazayeri, and D. Mandrioli. *Fundamentals of Software Engineering*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2002.

[49] E.E. Giladi and Y. Klar. When standards are wide of the mark: Nonselective superiority and inferiority biases in comparative judgments of objects and concepts. *Journal of Experimental Psychology: General*, 131(4):538–51, 2002.

[50] R.A. Guillemette. Usability in computer documentation design: Conceptual and methodological considerations. *IEEE Transactions on Professional Communication*, 32(4):217–229, 1989.

[51] J. Hartwick and H. Barki. Communication as a dimension of user participation. *IEEE Transactions on Professional Communication*, 44(1):21 –36, March 2001.

[52] M. Hertzum and A.M. Pejtersen. The information-seeking practices of engineers: Searching for documents as well as for people. *Information Processing & Management*, 36(5):761 – 778, 2000.

[53] T. Honderich, editor. *The Oxford Companion to Philosophy*. Oxford University Press, 2005.

[54] Darrel Ince, editor. *A Dictionary of the Internet*. Oxford University Press, 2009.

[55] P. Ingalls and T. Frever. Growing an agile culture from value seeds. In *Agile Conference, 2009. AGILE '09.*, pages 119 –124, August 2009.

[56] D. Jackson, M. Thomas, and L.I. Millett. *Software for Dependable Systems: Sufficient Evidence?* The National Academies Press, 2007.

[57] A. Jedlitschka, M. Ciolkowski, and D. Pfahl. Reporting experiments in software engineering. In F. Shull, J. Singer, and D.I.K. Sjøberg, editors, *Guide to Advanced Empirical Software Engineering*, pages 201–228. Springer London, 2008.

[58] M. Jorgensen and S. Grimstad. The impact of irrelevant and misleading information on software development effort estimates: A randomized controlled field experiment. *IEEE Transactions on Software Engineering*, 37(5):695–707, 2011.

[59] B.A. Kitchenham and S.L. Pfleeger. Personal opinion surveys. In Forrest Shull, Janice Singer, and Dag I. K. Sjøberg, editors, *Guide to Advanced Empirical Software Engineering*, pages 63–92. Springer London, 2008.

[60] B.A. Kitchenham, T. Dyba, and M. Jorgensen. Evidence-based software engineering. In *Proceedings of the 26th International Conference on Software Engineering*, ICSE '04, pages 273 – 281, May 2004.

[61] G. Kotonya and I. Sommerville. *Requirements Engineering - Processes and Techniques*. John Wiley & Sons, 1998.

[62] J. Law and E.A. Martin, editors. *A Dictionary of Law*. Oxford University Press, 2009.

[63] D. Leffingwell and D. Widrig. *Managing Software Requirements: A Unified Approach*. Addison-Wesley Professional, 2000.

[64] A.J. Lennon, B.C. Watson, C. Arlidge, and G. Fraine. 'You're a bad driver but I just made a mistake' : Attribution differences between the 'victims' and 'perpetrators' of scenario-based aggressive driving incidents. *Transportation Research Part F: Traffic Psychology and Behaviour*, 14(3):209–221, May 2011.

[65] J.-L. Lions. Ariane 5 flight 501 failure report by the Inquiry Board. `http://esamultimedia.esa.int/docs/esa-x-1819eng.pdf`, July 2012.

[66] S. Lohmann, S. Dietzold, P. Heim, and N. Heino. A web platform for social requirements engineering. In *Software Engineering (Workshops)*, volume 150, pages 309–315, 2009.

[67] K.M. Lord. *The Perils And Promise of Global Transparency: Why the Information Revolution May Not Lead to Security, Democracy, or Peace*. State University of New York Press, 2007.

[68] K. Lyytinen and R. Hirschheim. *Information systems failures: A survey and classification of the empirical literature*, pages 257–309. Oxford University Press, Inc., 1987.

[69] L.W. Mar, Y.-C. Wu, and H.C. Jiau. Recommending proper API code examples for documentation purpose. In *2011 18th Asia Pacific Software Engineering Conference (APSEC)*, pages 331–338. IEEE, 2011.

[70] C. McMillan, M. Grechanik, D. Poshyvanyk, Q. Xie, and C. Fu. Portfolio: Finding relevant functions and their usage. In *2011 33rd International Conference on Software Engineering (ICSE)*, pages 111–120. IEEE, 2011.

[71] H. Meijer, J.-H. Hoepman, B. Jacobs, and E. Poll. Computer security through correctness and transparency. In K. De Leeuw and J. Bergstra, editors, *The History of Information Security*, pages 637 – 653. Elsevier Science B.V., Amsterdam, 2007.

[72] A. Menéndez-Viso. Black and white transparency: Contradictions of a moral metaphor. *Ethics and Information Technology*, 11(2):155–162, 2009.

[73] R.T. Mercuri. Trusting in transparency. *Communications of the ACM*, 48(5):15–19, 2005.

[74] P. Meunier. Software transparency and purity. *Communications of the ACM*, 51 (2):104–104, 2008.

[75] Microsoft Corporation. Visualize transparency: Improving processes and services in government. `http://www.microsoft.com/industry/government/solutions/data_visualization/default.aspx`, May 2011.

[76] Microsoft Corporation. Distributed connectivity services. `http://msdn.microsoft.com/en-us/library/dd632026.aspx`, April 2013.

[77] Microsoft Corporation. Building a transparent process. `http://msdn.microsoft.com/en-nz/library/dd631988.aspx`, April 2013.

[78] P. Moles and N. Terry, editors. *The Handbook of International Financial Terms*. Oxford University Press, 1997.

[79] D. Moody. The "physics" of notations: Toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on Software Engineering*, 35(6):756 –779, 2009.

[80] T. Nakatani, T. Tsumaki, M. Tsuda, M. Inoki, S. Hori, and K. Katamine. Requirements maturation analysis by accessibility and stability. In *2011 18th Asia Pacific Software Engineering Conference (APSEC)*, pages 357–364. IEEE, 2011.

[81] G. Norman. Likert scales, levels of measurement and the "laws" of statistics. *Advances in Health Sciences Education*, 15(5):625–632, 2010.

[82] B. Nuseibeh and S.M. Easterbrook. Requirements engineering: A roadmap. In *ICSE - Future of SE Track*, pages 35–46, 2000.

[83] K. O'Hara and N. Shadbolt. Privacy on the data web. *Communications of the ACM*, 53(3):39–41, 2010.

[84] R.W. Oliver. *What is transparency?* McGraw-Hill, 2004.

[85] S. Patig. A practical guide to testing the understandability of notations. In *Proceedings of the fifth Asia-Pacific conference on Conceptual Modelling - Volume 79*, APCCM '08, pages 49–58, Darlinghurst, Australia, Australia, 2008. Australian Computer Society, Inc.

[86] N. Paul and A.S. Tanenbaum. Trustworthy voting: From machine to system. *Computer*, 42(5):23–29, 2009.

[87] M. Petre. Why looking isn't always seeing: Readership skills and graphical programming. *Communications of the ACM*, 38(6):33–44, 1995.

[88] W.G. Poole. The softer side of custom software development: Working with the other players. In *Proceedings. 16th Conference on Software Engineering Education and Training, 2003. (CSEE T 2003)*, pages 14–21, 2003.

[89] T. Punter, M. Ciolkowski, B. Freimut, and I. John. Conducting on-line surveys in software engineering. *International Symposium on Empirical Software Engineering*, 0:80, 2003.

[90] W. Reinhardt and S. Rinne. An architecture to support learning, awareness, and transparency in social software engineering. *iJET*, 5(S1):19–24, 2010.

[91] G. Rowe and L.J. Frewer. Public participation methods: A framework for evaluation. *Science, Technology, & Human Values*, 25(1):3–29, 2000.

[92] J. Rubin and D. Chisnell. *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. Wiley, 2008.

[93] H. Saiedian and R. Dale. Requirements engineering: Making the connection between the software developer and customer. *Information and Software Technology*, 42(6): 419 – 428, 2000.

[94] J.C. Sampaio do Prado Leite and C. Cappelli. Exploring i* characteristics that support software transparency. In *Proceedings of the 3rd International i* Workshop, CEUR Workshop Proceedings*, volume 322, pages 51–54, 2008.

[95] J.C. Sampaio do Prado Leite and C. Cappelli. Software transparency. *Business & Information Systems Engineering*, 2:127–139, 2010.

[96] A. Santana and D. Wood. Transparency and social responsibility issues for wikipedia. *Ethics and Information Technology*, 11:133–144, 2009.

[97] K. Schwaber and J. Sutherland. The scrum guide. `http://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum_Guide.pdf`, July 2012.

[98] M. Serrano and J.C. Sampaio do Prado Leite. Capturing transparency-related requirements patterns through argumentation. In *2011 First International Workshop on Requirements Patterns (RePa)*, pages 32 –41, August 2011.

[99] C.E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, January 2001.

[100] F. Shull and R.L. Feldmann. Building theories from multiple evidence sources. In F. Shull, J. Singer, and D.I.K. Sjøberg, editors, *Guide to Advanced Empirical Software Engineering*, pages 337–364. Springer London, 2008.

[101] M. Shuttleworth. Between subjects design. `http://explorable.com/between-subjects-design`, February 2013.

[102] D.I.K. Sjøberg, T. Dybå, B.C.D. Anda, and J.E. Hannay. Building theories in software engineering. In F. Shull, J. Singer, and D.I.K. Sjøberg, editors, *Guide to Advanced Empirical Software Engineering*, pages 312–336. Springer London, 2008.

[103] F. Small. Ministerial inquiry into INCIS. `http://www.justice.govt.nz/publications/global-publications/m/ministerial-inquiry-into-incis`, July 2012.

[104] Software Transparency Team. Software transparency. `http://transparencia.les.inf.puc-rio.br/english/eng_index.html`, December 2009.

[105] I. Sommerville and P. Sawyer. *Requirements Engineering: A Good Practice Guide*. John Wiley & Sons, Inc., New York, NY, USA, 1997.

[106] A. Stevenson, editor. *Oxford Dictionary of English*. Oxford University Press, 2010.

[107] D. Stoljar. Transparency. In T. Bayne, A. Cleeremans, and P. Wilken, editors, *The Oxford Companion to Consciousness*. Oxford University Press, 2009.

[108] R. Stroud. Transparency and reflection in distributed systems. In *EW 5: Proceedings of the 5th workshop on ACM SIGOPS European workshop*, pages 1–5, New York, NY, USA, 1992. ACM.

[109] M. Svahnberg, T. Gorschek, M. Eriksson, A. Borg, K. Sandahl, J. Borster, and A. Loconsole. Perspectives on requirements understandability–for whom does

the teacher's bell toll? In *Requirements Engineering Education and Training. REET'08.*, pages 22–29. IEEE, 2008.

[110] B.C.Y. Tan, H.J. Smith, M. Keil, and R. Montealegre. Reporting bad news about software projects: Impact of organizational climate and information asymmetry in an individualistic and a collectivistic culture. *IEEE Transactions on Engineering Management*, 50(1):64 – 77, February 2003.

[111] C. Tenopir and D.W. King. *Communication patterns of engineers.* Wiley-IEEE Press, 2004.

[112] Transparency International. Frequently asked questions about corruption. `http://www.transparency.org/news_room/faq/corruption_faq`, January 2010.

[113] W. Trochim. *Research methods knowledge base.* Atomic Dog Pub., Cincinnati, OH, 2001.

[114] Y.-C. Tu, C. Thomborson, and E. Tempero. Illusions and perceptions of transparency in software engineering. In *2011 18th Asia Pacific Software Engineering Conference (APSEC)*, pages 365–372. IEEE, 2011.

[115] M. Turilli and L. Floridi. The ethics of information transparency. *Ethics and Information Technology*, 11(2):105–112, 2009.

[116] A. Vaccaro and P. Madsen. Transparency in business and society: Introduction to the special issue. *Ethics and Information Technology*, 11(2):101–103, 2009.

[117] A. Vaccaro and P. Madsen. Corporate dynamic transparency: The new ICT-driven ethics? *Ethics and Information Technology*, 11(2):113–122, 2009.

[118] J.M. Verner, S.P. Overmyer, and K.W. McCain. In the 25 years since the mythical man-month what have we learned about project management? *Information and Software Technology*, 41(14):1021 – 1026, 1999.

[119] D.B. Walz, J.J. Elam, and B. Curtis. Inside a software design team: Knowledge acquisition, sharing, and integration. *Communications of the ACM*, 36(10):63–77, 1993.

[120] D.J. Weitzner, H. Abelson, T. Berners-Lee, J. Feigenbaum, J. Hendler, and G.J. Sussman. Information accountability. *Communications of the ACM*, 51(6):82–87, June 2008.

[121] C. Williams, P. Wagstrom, K. Ehrlich, D. Gabriel, T. Klinger, J. Martino, and P. Tarr. Supporting enterprise stakeholders in software projects. In *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering*, pages 109–112. ACM, 2010.

[122] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering: An Introduction*. Kluwer Academic Publishers, Norwell, MA, USA, 2000.

[123] M. Zaki and P. Forbrig. User-oriented accessibility patterns for smart environments. In *Human-Computer Interaction. Design and Development Approaches*, pages 319–327. Springer, 2011.

[124] M.V. Zelkowitz and D.R. Wallace. Experimental models for validating technology. *Computer*, 31(5):23–31, 1998.