A Field Study of Domain Knowledge Sharing in the Software Development Industry in New Zealand

Christian Harsana Ekadharmawan

A thesis submitted to
Auckland University of Technology
in partial fulfilment of the requirements for the degree of
Master of Computer and Information Sciences (MCIS)

2008

School of Computer and Mathematical Sciences

Primary Supervisor: Jim Buchan
Secondary Supervisor: Prof. Stephen MacDonell

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

ABBREVIATIONS

AUT	Auckland University of Technology
BA	Business Analyst
CEO	Chief Executive Officer
CFO	Chief Financial Officer
NZ	New Zealand
IS	Information System
IT	Information Technology
OO	Object Oriented
RE	Requirements Engineering
SME	Small Medium Enterprises
UA	User Acceptance Test
UML	Unified Modelling Language

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Christian Harsana Ekadharmawan

14 October 2008

Acknowledgement

This research is one of the biggest academic achievements in my life and I am very grateful for the opportunity to conduct this research as well as the opportunity to learn about requirements engineering practice in real world settings. I am aware that it is impossible for me to reach this stage without support from many people. Therefore, I would like to make the most of this opportunity to show my gratitude to those people.

First and foremost, I would like to thank my primary supervisor, Jim Buchan, particularly for his ideas, suggestions, knowledge, encouragements, time, and his efforts in guiding me to be a better researcher and writer. I also wish to thank my second supervisor, Prof. Stephen MacDonell for his ideas, suggestions, and encouragements as well.

Special thanks are extended to all of the participants of this research for their warm welcome, valuable time, and their kind assistance, since without them the research would not have been possible.

I would like to thank people from AUT, particularly Krassie Petrova for her kind help and support, as well as my colleagues: Pat, Rafiq, Omkar and Wilford for their help, support, and encouragement during the development of this thesis.

My gratitude is also extended to my friends from St. Andrews Youth Group, for their prayers, support and encouragement when I was feeling discouraged.

I would like to thank my family, for their continuous support, love, care and countless prayers during my study here and during the making of this thesis.

I also would like to thank Josephin, for her love, affection, support, care, and countless prayers.

Last but not least, I would like to thank God for His love, His patience, His guidance, His providence, and ultimately who makes all of these things possible. I hope this piece of work can be used for His glory. To God be the glory. Amen.

This research was approved by the Auckland University of Technology Ethics Committee on 28 September 2007, AUTEC Reference number 07/170.

Abstract

In contemporary software development, an emergent understanding of the problem domain and envisioned goals forms the basis of designing, testing and development activities. Lack of a common understanding of the domain can result in costly rework or client dissatisfaction. Research shows that the development of shared understanding in this context is a complex and error-prone process and there is room for improvement. Is this because practitioners are not following suggested practice from literature? Or are the actual barriers to shared understanding not being addressed by current tools and techniques? Is the development of shared domain understanding even viewed as problematic (or even important) by practitioners? These are some questions that need to be investigated in order to effectively design process improvements and tool support in this area, yet there is little information related to this.

This study takes a multi-case study approach, which incorporate semi-structured interviews with representative from ten small-to-medium organisations. This study focuses on the vendor's perspective and includes a mix of application-domains. Result of the interviews is analysed to discover themes and patterns related to an analysis-framework constructed from the literature review.

The findings indicate that vendors perceive the process of developing shared application-domain understanding with their clients as being both problematic and important to a successful implementation. Twelve barriers have been identified from the analysis. The results also confirm that the process of sharing understanding development is generally perceived as being evolutionary and collaborative. This process is described by most interviewees comprises iterative phases of elicitation, confirmation and refinement of the understanding. A definite preference for face-to-face interaction is evident at regular times throughout development, particularly in early stages, although the importance of ad-hoc communications by phone or email, as domain knowledge needs arise, is also emphasised. Access to cooperative domain-expert throughout development is generally seen as a critical success factor. Several companies report using in-house domain-expert as client "proxies" in this regard. There is a mix of attitudes apparent regarding the direct

communications of developers with client stakeholders. This ranged from insisting that developers are involved from initial elicitation and “kick-off” meetings, to “shielding” developers almost entirely from client. In terms of representations of understanding, participants relate natural-language, screen-shots, mock-ups, prototypes and product-demonstrations as the most useful artefacts for sharing and confirming understanding of the problem domain. They emphasise the importance of flexibility and client familiarity with the representations. In general, there is no clear separation between problem and solution spaces evident when the interviewees discussed representations of understanding, and the preference seems to be for concrete rather than abstract representations.

In conclusion, comparisons between the findings and literature generally confirm contemporary thinking regarding domain knowledge sharing, although a number of barriers were given particular emphasis in this field study. The use of computer-based tool support is not widespread and the need to improve the domain knowledge sharing process and tool support in practice is widely acknowledged by the participants in this investigation. This study has identified some fruitful areas of research in this regard.

Chapter 1

Introduction

1.1. Introduction

Requirements Engineering (RE) is one of the central collaborative processes that take place in developing a software system. It is well documented that failure in developing good (e.g. sound, justified, clear) requirements can affect the downstream development phases and can contribute to the failure of a project (Emam & Madhavji, 1995; Flynn & Jazi, 1998; Kudikyala & Vaughn, 2005; Lawrence, 1998; Rosencrance, 2007). This argument is further supported in a report by Standish Group which states that in the USA between 30% and 48% of IT projects fail because of requirements-related problems (Hilton, 2003).

A number of researchers in the area of requirements engineering (for example Alcázar and Monzón, 2001; Offen, 2002; Kotonya & Sommervile, 2002, as cited in Tuunanen, 2003) note that one of the success factors for developing good requirements, is that the stakeholders, which refers to actors on both the supplier (vendor) and the client sides, share some level of understanding of the "problem" domain within which the "solution" system will operate. This is related to the need for stakeholders related to the project to have at least a similar vision of the desired goals, business process, etc. (Gordijn, Yu, & Van der Raadt, 2006; Nuseibeh & Easterbrook, 2000; Segal, 2005).

This common understanding of the problem domain will develop if there are knowledge sharing activities occurring between the vendor and the client groups. In the context of software development activities, these knowledge sharing activities are most intense in the RE activities, especially during the elicitation process (Nuseibeh & Easterbrook, 2000). According to Deshpande, de Vries and van Leeuwen (2005), development of shared understanding of a belief requires a continual process of reshaping the understanding through communication, which, in terms of RE is reflected through a series of validation and verification activities regarding the shared knowledge. These arguments demonstrate that there is a close relationship between RE activities and the development of shared understanding.

However, developing shared understanding is not an easy task to undertake, as it is communication intensive (Coughlan, Lycett, & Macredie, 2003), which is often problematic over extended periods of time (Al-Rawas & Easterbrook, 1996). Alcázar & Monzón (2000) as well as Cao & Ramesh (2008) and Hsieh (2006) note some of the factors which contribute to the difficulty of sharing understanding between two such groups with different backgrounds (client and vendor). This includes factors such as different world views, mental models, culture, values, interests, and knowledge. Segal (2005) points out that inadequate technology, representations, or artefacts to support sharing of understanding can also hinder its effective development.

Evidence showing difficulties in developing shared understanding and its numerous problems has provided the motivation to conduct this research, with the aim of identifying and gaining deeper understanding of the problem areas that need to be improved or even changed, as well as aspects of sharing understanding that are effective.

Despite the clear recognition in literature that sharing domain understanding is an important factor of successful software development, there seems to be little research in the software engineering literature that focuses on understanding and supporting the development of shared understanding of a problem domain between the client and vendor group, especially in the context of a study of industry based practice. There are some studies on shared understanding that investigate specific dimensions of shared understanding rather than a broader scope. For example: Aranda, Khuwaja and Easterbrook (2007b) investigate the dynamics of shared understanding, while Hsieh (2006) focuses specifically on the relationship between shared understanding and culture. Both of these studies focus their research in the context of development team members only. In addition, there are other studies which focus on shared understanding in the context of virtual teams (Hinds & Weisband, 2003; Rooij, Verburg, Andriessen, and Hartog, 2007). The emphasis of this thesis is more on understanding the experience and practice of developing shared understanding in a commercial software development environment. Some other existing studies on RE focus on requirements development itself, for example: studies from Cao & Ramesh (2008) Emam & Madhavji (1995), and Nikula, Sajeniemi, & Kalvianen (2000). Research from Neil & Laplante (2003) investigates the state of current practice of RE, and their findings provide insight on the current tools and techniques used in practice. However the work does not provide a clear description of how stakeholders ensure the development of shared understanding of the problem domain – if they do at all. Another related study by Segal (2005), involves a single case study of a software development project. One of the findings is the insufficiency of project documentation for developing shared understanding between a scientist group (client) and developers. However, this study does not put development of shared understanding at the centre of the research.

1.2. Research goals and aims

In order to improve the current practice of developing shared understanding of the problem domain, a better understanding of the phenomenon is required to help the researchers determine what aspects they should devote their attention to. Therefore, the goal of this research can be defined as:

> *To understand the process of developing shared understanding of the problem domain. This is expected to provide insights that are useful for improving current practice or suggesting potential fruitful areas of research.*

In order to achieve such a goal, reviews of practices that are believed to promote development of shared understanding development are required. The selection of the practice is limited to RE, where the problem domain is being shared throughout most of the activities (Nuseibeh & Easterbrook, 2000). Therefore, the aim of this research can be defined as:

> *To explore current RE practices that influence the of development of shared understanding of the problem domain.*

1.3. Research objectives

The aim of exploring current RE methods, prompts the establishment of several research objectives. The first main research objective is dealing with the investigation and discussion of the communication process between the vendor and client group, which is usually represented by the purchaser of a software product. This particular research objective is based on the idea that developing shared understanding is achieved through communication between the two groups, as implied by Deshpande et al. (2005). Therefore the first research objective of this research is:

> *To investigate and analyze the communication process between the vendor and the client group in terms of developing shared understanding of the problem domain.*

However, the objectives of investigating the communication process seem to be broad and need to be specified. Therefore, areas of communication that seem most important (largest influence) are emphasised during the investigation. This includes the nature of the communication activities, their frequency, who is involved in the activities, the tools and techniques being used to support the activities, the efficacy of the activities, as well as the kinds of knowledge representations used. Consequently, the research objectives are expanded to:

> *To identify and analyze the activities that are relevant to shared understanding development, including their frequency and who is involved in them.*

> *To identify tools and techniques used that potentially support the development of shared understanding, as well as their effectiveness.*

To identify the types of representations of problem domain understanding that are used in the RE process.

The second main research objective is dealing with what the practitioners' perceptions and experiences are regarding developing shared understanding. This idea is based on the notion that the goal of obtaining insights for improving the development of shared understanding practice are likely to be achieved in practice only if the practitioners appreciate the difficulty and value of developing shared understanding. Consequently, adequate appreciation of shared understanding development should lead to adequate allocation of practitioners' efforts and desires to improve the process and identify problems, difficulties, or barriers hindering the development of shared understanding. Therefore some additional objectives of this research are:

To investigate practitioners' perceptions of the importance of the development of shared understanding, as well as their effort in achieving such a condition.

To identify barriers that would hinder practitioners from developing shared understanding, as well as their methods of overcoming these obstacles.

Additionally, as is highlighted by Glass (2007) and Nikula et al. (2000), there is frequently a gap between what is reported through literature and what actually occurs in practice. This research opens an opportunity to compare what is being reported in literature with the current practice of requirement engineering, as well as investigating the reasons for any differences. Therefore an additional research objective is added to this research:

To compare what is being reported in literature and current practice, in order to identify any gaps, and possible explanations for it.

1.4. Research questions

In order to focus this research, research questions have been established in line with the research objectives described earlier. The research questions for this thesis are:

Research Question 1:

Do stakeholders perceive developing shared understanding as problematic (or difficult)?

Research Question 1a

If so, what do they say the barriers are?

Research Question 2:

What do stakeholders say they do (processes, techniques, tools, representations) in order to develop shared understanding of the problem domain with their client?

Research Question 3:

How does what the stakeholders say they do compare to what is reported in research literature?

The answers to these research questions will be covered in chapter 4.

1.5. Research contributions

This research is contributing to evidence-based research in the area of RE practice and should be of value to both the software engineering research community and practitioners.

For the research community, firstly, apart from contributing to the existing body of knowledge on how shared understanding is developed in practice, this study may uncover some of the causes of the research-practice gap. This could provide significant insights for the research community in order to improve the research applicability and uptake by practitioners. Secondly, identification of recurring patterns of the barriers in developing shared understanding across a variety of problem domains would also provide the research community some insights into areas that need to be improved or undeveloped research areas.

As for practitioners, the findings regarding the shared development practice may provide them with points for reflecting upon their own practices, new ideas on how to improve their current practices, or even provide them with a warning regarding particular practices in certain contexts. All of these things can potentially be used to improve the current practice of practitioners.

1.6. Research scope

This research investigates the development of shared understanding practices in software development projects in New Zealand. In order to do so, this research is focused on organizations whose main business is software development, both bespoke development or product-driven development.

In terms of geographical area, participant organizations are selected from the greater Auckland region, New Zealand. This reason for this is mainly pragmatic: easy access to the organisations by the researcher.

The research was initially intended to investigate the phenomenon of shared understanding development from both vendors' and the clients' perspectives, in order to compare them and gain further insights from the two perspectives. It was decided that since there was a relatively high degree of uncertainty regarding the time needed for the field work, subsequent interview transcriptions and data analysis, that the study would be split in two phases, phase one being the vendors' viewpoints and phase two investigating the clients' perspectives. If phase one took longer than initially planned then phase two could be left as future research, but there would be enough data for a significant analysis from phase one. It turns out that it has proved unworkable to complete both phases in the time available and only the vendors' viewpoints have been investigated. Therefore, in order to mitigate the risk, the research is divided into two phases, with the first phase focussing on the vendor side and the second phase focussing on the client side. As for the latter, it is intended to be a future research.

There are also several reasons why the research on the vendor side was conducted ahead of the client side:

- Shared understanding of problem domain tends to be more of a burden to the vendor side, as the client side is usually more familiar with the problem domain.
- In the practice, it seems that development of shared understanding is driven more by vendor side than the client side. Thus, experience on the processes, tools and techniques used in developing shared understanding are more likely to come from the vendor side.
- In term of the readers who presumably would be IT practitioners, vendors' experience is more likely to provide knowledge that is applicable in the practice.

For these reasons, investigations on the clients' perceptions will be regarded as an extension of this study for future research.

1.7. Research methodology

The research is considered exploratory, considering the aim of this research and the limited knowledge available regarding the development of shared understanding in the context of RE practices.

In term of epistemology, an interpretivist approach has been selected, since it is the peoples' perceptions and experience of the phenomenon of shared understanding development that is of interest.

This interest in collecting peoples' perceptions of their experience of a phenomenon suggests a qualitative research approach and a qualitative data collection. As discussed in more detail in Chapter three, this influenced the selection of multiple-case study as the qualitative research method used.

In terms of data collection methods, semi-structured interview is preferred over structured interview or direct observation, because of its flexibility and ability to combine open questioning with further probing to gain deeper understanding of the phenomenon. The research methodology and methods are discussed and justified in more detail in Chapter three.

1.8. Thesis outline

As the final report of this research, this thesis consists of five chapters:

- ***Chapter 1 – Introduction***: introduces the background of the topic and research rationale for conducting this research.
- ***Chapter 2 – Literature review***: provides a review of past works that are relevant to this research on phenomenon of shared understanding development.
- ***Chapter 3 – Research methodology***: discusses and justifies the research methodology and the research methods employed in conducting this research.
- ***Chapter 4 – Research findings and discussion***: presents a summary of the findings, analysis of the findings, as well as the explicitly answering the research questions.
- ***Chapter 5 – Conclusion***: presents the conclusion of the research, limitations of this study, as well as potential future research.

1.9. Chapter summary

This chapter provides the reader with the background and rationale of this research as well as its goal, aim, and objectives. The research objectives described provide a basis for formulating the research questions. This chapter also covers the contributions that this research makes to the practitioner and research communities, its scope and a brief introduction on the research methodology of this research. Finally, an outline of the thesis structure is presented.

Chapter 2

Literature Review

2.1. Introduction

This chapter summarises related current theory and practice from research literature and provides a wider context and motivation for this study. Firstly, in section 2.2, the general concepts of 'shared understanding' and the 'problem domain' are explored and working definitions of these concepts are provided for this study. The section continues by presenting an evidenced case for the importance of shared problem domain understanding in the context of RE activities. This justifies the value in continued research into this area. The next section provides the definition of 'stakeholder' used in this study, which includes both vendor and client groups. Next a model is presented that summarises the current understanding of stakeholder' possible "roles" and interactions in the process of developing shared domain understanding. This helps to understand the nature of who is involved in sharing their understanding and the significance of this to the research questions. Section 2.4 provides some insights into the current state of practice of developing shared understanding in the software development context by reviewing empirical studies in this area. This reviews knowledge about the activities, techniques and supporting tools that are employed by practitioners to develop shared domain understanding. In Section 2.5 the significance of the representation of understanding or knowledge to sharing that understanding is explored.

Throughout this chapter there is an underlying thread, interwoven with the subject of each, section that investigates what literature says the barriers to developing shared understanding are.

In the conclusion of this chapter, the various factors that may influence the development of shared domain understanding, identified from the literature review, are synthesised and presented in a simple model that relates to the research questions. This model is subsequently used as a framework for data gathering and analysis.

2.2. The context and importance of shared understanding

As noted by Cannon-Bowers and Salas (2001), the word "shared" in the context of collaborative activities between individuals can be defined in a number of ways (see table 2.1). These definitions are used to form a working definition of "shared understanding" in the context of this study. Moreover, an examination on "problem domain" (as another significant element of this study's context) is also presented to improve the clarity of this study's purpose. In addition, to provide a wider context to the practice of developing shared understanding, its relationship with requirements engineering practice is examined and covered in this section as well. Lastly, to strengthen the justification and value of this study, a review on evidences that indicate the importance of developing shared understanding is presented at the end of this section.

2.2.1. The notion of 'shared understanding'

From the literature reviewed, several attempts at defining shared understanding are identified. Ottenheijm, Van Genuchten and Geurts (1998) define 'shared understanding of a problem' as: 'shared understanding of all the available information, value trade-offs and alternatives of the problem'. The major flaw of this definition is certainly the absence of an explanation on the nature of shared understanding itself, as it only emphasises the items that need to be shared. A more thorough definition comes from Deshpande et al. (2005) who define shared understanding as: 'an objected state achieved through interactive process by which a common ground between individuals in constructed and maintained'. This common ground emphasises information such as the mutual knowledge of beliefs and assumptions. This seems a fuller definition compared to the previous one because it covers the intended result, as well as the nature of the (interactive) process for developing shared understanding. However, little is said about the meaning of “shared”.

Cohen and Gibson (2003), in their definition, do define what is being shared. They define shared understanding as: ‘The degree of cognitive overlap and commonality in beliefs, expectations, and perceptions about a given target’. They go on to discuss that what is shared is the goals, work and group processes, tasks, and what each team member brings to the team task (e.g. member knowledge, skills, and abilities. The importance of defining ‘shared’ in this context is consistent with Cannon-Bowers and Salas’ (2001) recommendation of defining ‘shared’ when dealing with this kind of topic in order to increase the clarity of the study. Cohen and Gibson’s (2003) definition of ‘shared’ closely corresponds to the ‘overlapping’ definition of Cannon-Bowers and Salas’ (2001) (see table 2.1).

Meaning	Description
Shared or overlapping	Refers to a situation where two or more persons need to have some common base knowledge related to a task.
Similar or identical	Refers to a situation where two or more persons hold similar attitudes or belief in order to draw common interpretations.
Compatible or complementary	Refers to a situation where knowledge leads individuals to have similar expectations.
Distributed	Refers to a situation in which knowledge is distributed between members, thus coordination is necessary to utilize the knowledge.

Table 2.1 - Categories of what “shared” means (Cannon-Bowers & Salas, 2001)

In the context of software development project, Cohen and Gibson’s (2006) definition seems the most complete and will be used as the meaning of ‘shared understanding’ throughout this thesis.

In exploring the concept of ‘shared problem domain understanding’ the meaning of ‘problem domain’ needs to be clearly defined also in the context of this study. A working definition of this concept is developed in the next section.

2.2.2. The notion of 'problem domain'

In order to increase the clarity of the study on shared understanding, Cannon-Bowers and Salas' (2001) also recommend defining the knowledge that is intended to be shared. In the context of a software development project, knowledge of goals and business imperatives are certainly important, since they provide the directions and boundaries for a project. However, since a software-development project is typically done to solve a (business) problem (or realise an opportunity), understanding the problem and its characteristics is crucial in order to avoid developing something that is unwanted or unnecessary (i.e. addressing the "wrong" problem). A problem and its related properties are commonly known as a problem domain. Hall, Jackson, Laney, Nuseibeh, and Rapanotti (2002a) pragmatically define a problem domain as 'problems located in the real world that need a solution'.

In the literature the terms 'domain knowledge' and 'application domain' are often used as synonyms for 'problem domain' and are often contrasted with the term 'solution domain'. For example, Hammond, Rawlings, & Hall, (2001), describe 'application domain' as 'aspects of the real world that are relevant and need to be improved', very similar to the previous definition of 'problem domain'. (Hammond et al., 2001) distinguish the term 'domain knowledge' as properties of the real world that are known (or assumed) to be true. Figure 2.1 depicts the relationships between these terms that are used throughout this thesis.

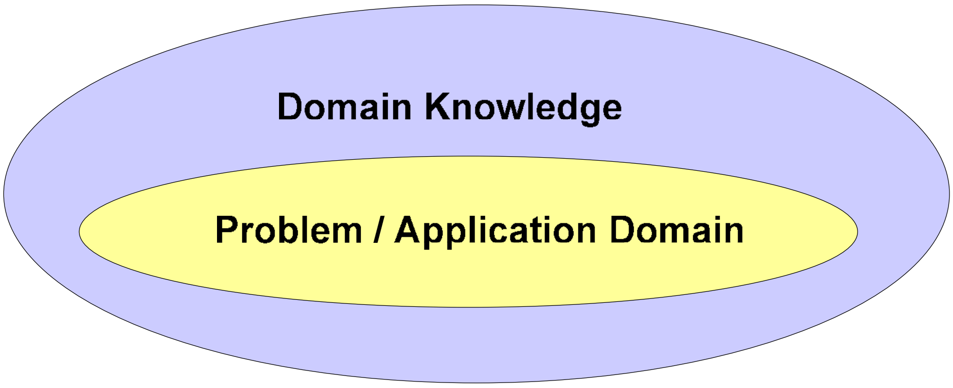

Figure 2.1 – Relationship between domain knowledge and problem/application domain

In this interpretation, then, the relationship between domain knowledge and problem domain implies that, in order to understand a problem domain, understanding of domain knowledge related to the problem is needed also.

The idea of a problem domain is often contrasted with the notion of a solution domain in literature (e.g. Alcázar & Monzón, 2001, Hall et al., 2002a; Hammond, et al., 2001; Monarchi & Puhr, 1992) The solution domain is defined as properties of the real world that are wished to be true (with the help of a product that need to be built). This is closely related to the common use of the concept of 'requirements' in the sense that requirements often capture envisioned changes to the current properties of the real world.

Having clarified the meaning of shared problem domain understanding, its significance in the context of d requirements engineering is discussed in the next section.

2.2.3. Relationship with Requirements Engineering

As noted in chapter 1, RE is a collaborative activity involving the vendor and the client groups throughout a software engineering project. RE involves activities that are intended to develop a shared understanding of the requirements that may then be implemented.

Literature in the RE area indicates that the development of shared understanding of the problem domain is fundamental to many RE activities. One view of this relationship is depicted in figure 2.2.

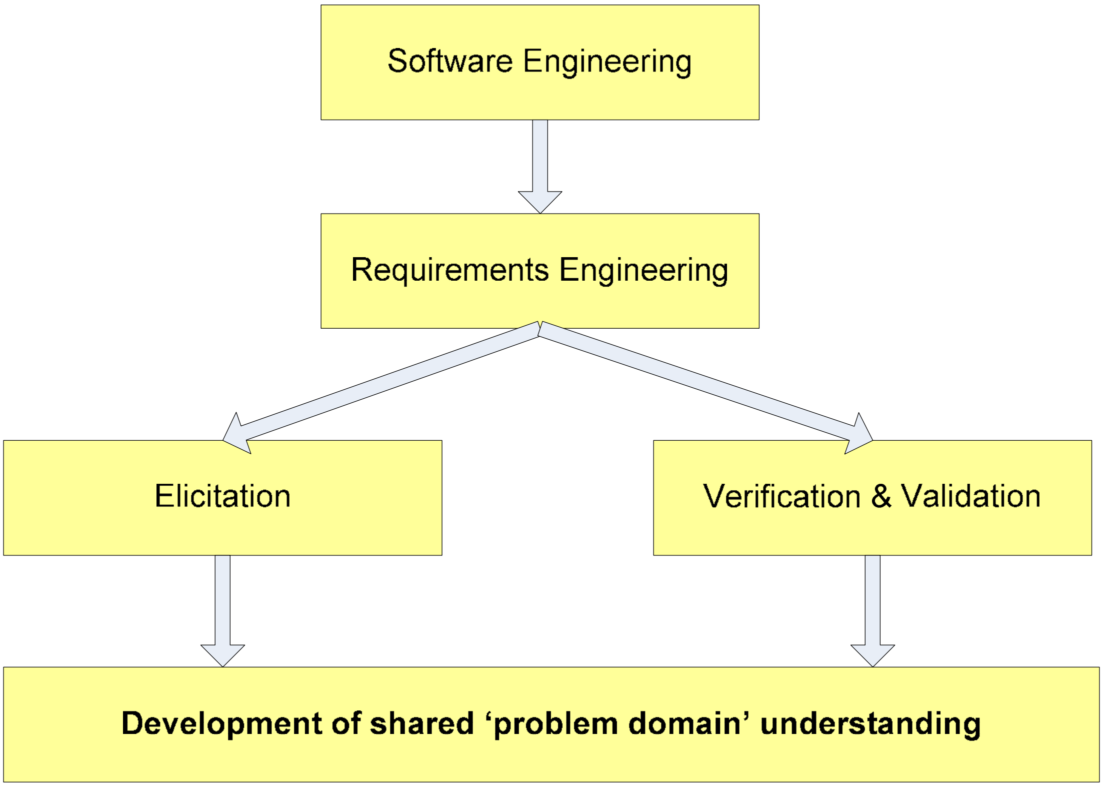
Software Engineering
Requirements Engineering
Elicitation
Verification & Validation
Development of shared 'problem domain' understanding

Figure 2.2 – Development of shared 'problem domain' understanding in software project

According to Kotonya and Sommerville (2002 as cited in Tuunanen, 2003), application domain knowledge is "embedded" in software requirements, therefore developing shared understanding of the requirements, which is considered as one of the main tasks of RE practice, means developing shared understanding of the application domain as well. A similar conclusion is also expressed by Alcázar and Monzón (2001), who argue the importance of the problem domain as a context for the software requirements and that an adequate level of shared understanding of the requirements implies a shared understanding of the problem domain. They show that one of the key opportunities for achieving this shared understanding is during the requirements elicitation activities.

In a study by Jarke (1998), he identifies that the development of shared understanding of issues related to a system (i.e. the problem domain) is one of the key task dimensions of RE practice. Several other studies indicate that understanding of the application domain/domain knowledge/problem domain is one of the foundations for a successful system development project (e.g. Alcázar & Monzón (2001), Hammond et al., 2001; Offen, 2002).

Some of the specific RE activities associated with development of shared domain understanding are identified by Nuseibeh & Easterbrook (2000) as elicitation, negotiation, verification, and validation of requirements. Alcázar and Monzón (2001) and Offen (2002), show that modelling and analysis, which are both also regarded as RE activities play an important role in developing understanding of the problem domain. . In addition Damian (2007) also argues that knowledge validation (which is an RE practice) is one mechanism for reshaping shared understanding of shared knowledge.

The conclusion then, from reviewing theory and practice of RE in the literature, is clearly that RE activities are closely linked with the development of shared understanding of the problem domain. The next section extends this notion to a more causal relationship between effective development of shared problem domain understanding and the efficacy of the software development process and quality of the software product.

2.2.4. Why is it important to develop shared understanding?

Various studies suggest that a high level of shared understanding brings a positive effect to a project in general (e.g. Aranda et al., 2007b; Cannon-Bowers & Salas, 2001; Hinds & Weisband, 2003; Hsieh, 2006). Considering this in more detail, Alcázar and Monzón (2001) argue that a problem domain provides context for any requirements established based on that particular domain. Situating the requirements into the problem domain would be useful for developing a common or shared understanding of the problem. Having a shared understanding of the problem domain would certainly lower the possibilities of changing and unclear requirements (Curtis et al., 1988; Rickman, 2001), which are evidenced as contributing to project failure (see or example Standish Group, 1995 as cited in Lawrence, 1998; and Rosencrance, 2007). In turn, clear and stable requirements would minimize the risk of (late) errors at implementation, and ultimately would contribute to the quality of the product itself (Coughlan & Macredie, 2002; Hammond et al., 2001; Hinds & Weisband, 2003). This also has significant cost implications , since Leffingwell and Widrig (2000, as cited from Hammond et al., 2001) plausibly calculate that the cost for finding the source of errors during the maintenance stage is around 200 times more than during the requirements engineering stage, when the problem domain understanding and sharing is generally most intense. So this is an argument for putting early and repeated effort into understanding and sharing the domain knowledge.

A high level of shared understanding of a problem domain is also important for ensuring that the development process is aimed at fulfilling the needs or solving the problems of individuals in that domain. Consequently, this would increase the degree of the end-product's usefulness and usability as well (Coughlan & Macredie, 2002). As a result, it ensures and might even increase the client group's satisfaction as well.

Additionally, having a high level of shared understanding of a problem domain provides a common base for interpreting requirements (Alcázar & Monzón, 2001) that could ensure that both the development group and client group would at least have similar perspectives of an issue. It could be argued that this should improve the cohesiveness of both groups, which Cannon-Bowers & Salas (2001) argue will lead to higher levels of trust among the stakeholders. This in turn will provide support of consensus building among the stakeholders which, according to Cannon-Bowers & Salas, would increase project productivity and increase the likelihood that the project is completed on time (Rickman, 2001).

In summary, developing shared understanding of the problem domain is important because it can have a positive effect on:

- the likelihood of changing or unclear requirements,
- the risk of late errors at implementation,
- the client group's satisfaction,
- the project productivity,
- the quality of the final system developed.

Having established in this section that the quality of shared domain understanding can have a significant influence on the effectiveness and efficiency of RE activities and the quality of the final software product, the next sections investigate some of the factors that literature shows may influence the quality and depth of this shared understanding and its development. The actors involved in refining and sharing their understanding are clearly central to the development and quality of the understanding. Their levels of involvement, commitment and knowledge, as well as values, learning styles and previous experiences and are some of the factors that can influence the degree of understanding shared. The next section surveys literature regarding the roles the stakeholders play in the interactions and some of their characteristics that influence development of shared domain understanding. The meaning of 'stakeholder' as used in this study is also defined and some justification for this definition presented.

2.3. Stakeholders in development of shared understanding

It is widely accepted that the accurate identification of stakeholders' needs can significantly affect the quality of a developed software product (Aurum & Wohlin, 2006). In the software RE context, the majority of the research community describes stakeholders as people who influence the requirements of the software, or people who are impacted by the software itself (for example see: Alexander & Robertson, 2004; Aurum & Wohlin, 2006; Glinz & Wieringa, 2007). Quite often the word "stakeholder" is used to only refer to the client side. However in this study, the term stakeholders refers to everyone who influences or get influenced by a project, with a particular emphasis on the distinction between stakeholders from the client side and the developer (software supplier or vendor) side.

Typically, as stated by Nuseibeh and Easterbrook (2000), the stakeholders from the client group involved directly in sharing their understanding of the problem domain could include:

- Customers or clients who sponsor or provide financial support in exchange for a software project's product, typically senior management, procurement managers, or members of the Board.
- Personnel who are directly responsible for the business process(s) that the software supports, generally line managers.
- In-house development, technical or project management roles.
- End-users who interact with the software as part of their day-to-day work.

The client-side stakeholders may extend to the customers, suppliers, distributors, competitors and regulators of the direct client group, in terms of the need for sharing understanding and knowledge with the vendor stakeholders.

The vendor stakeholder group with a need to share understanding of the problem domain with the client stakeholders include:

- Various roles in the development team who are responsible for the design, construction, testing, installation and maintenance of the software.
- Various analysts who are responsible in some sense for bridging the “gap” between the client stakeholders and the more technical vendor stakeholders and communicating this understanding to the development team (Damian, 2007).
- Senior management who are responsible negotiating terms of the client-vendor
- Sales and marketing personnel, often first point of contact (and sometimes continued contact) with the clients
- Distributors of the software product or service

In order to achieve shared understanding of the problem domain, there is a process of knowledge acquisition and knowledge sharing involving interactions between those stakeholders from client side and developer side (Damian, 2007).

In addition there are similar knowledge sharing interactions *within* each stakeholder group, since the domain understanding often needs to be developed with a wider group than those involved in direct vendor-client interactions. This intra-stakeholder group shared understanding is not the focus of this study, but rather it is the inter-stakeholder understanding between the client and vendor stakeholder groups that is studied. This is motivated by studies such those by Saiedian & Dalel (2000) and Segal (2005) who emphasise the complex nature and importance of these problematic inter-group interactions.

To gain an understanding of the complexity and issues associated with developing vendor-client shared understanding of the problem domain needs, it is useful to firstly explore the main roles of the participating stakeholders involved in this process.

2.3.1. Roles of stakeholders in developing shared understanding

In the context of developing shared understanding between client and developer stakeholder groups, various stakeholder roles can be identified, as shown in figure 2.3. It is important to note that some of these roles are not always present in a project because of various factors such as methodology applied by the organizations, the organization's structure, complexity of the project, etc.

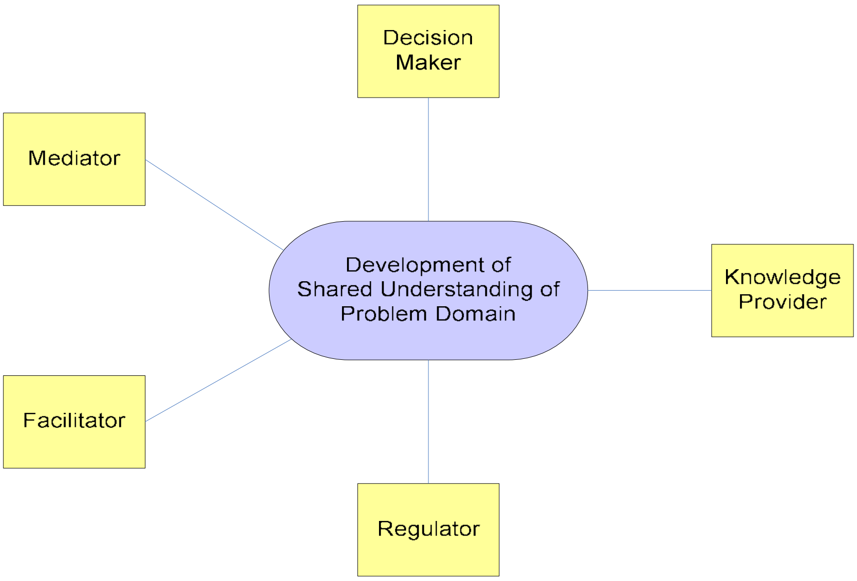

Figure 2.3 – Stakeholders' roles in development of shared understanding

This model is the result of a synthesis of literature from requirements engineering research, communications research and requirements elicitation theory. The next few sections describe the roles in more detail and discuss which roles may have the most impact on developing vendor-client shared domain understanding.

2.3.1.1. Knowledge provider

Knowledge providers are the stakeholders who have the knowledge that is relevant to the product that is being developed. This may include knowledge such as client stakeholders' needs, goals, vision, application domain knowledge and may involve sharing both explicit and tacit knowledge. In a development project, this particular role is usually seen in:

- Users or potential users (in terms of market driven product) who typically provide the functional requirements and tacit knowledge is often elicited from them. Moreover, they usually provide feedback on the product's usability as well.
- The business strategists, who provide high level goals and requirements.
- Project managers who provide the priorities of the client organization, their standards and their policies (Damian, 2007).
- Project sponsors (the owner of the business or the head of department) who may provide the vision and goal of the project.
- And in the case of market-driven or commercial off-the-shelf (COTS) products, account managers, marketers or sales representatives who tend to take the role of the provider of knowledge provider as well (Hofmann & Lehner, 2001).

Such knowledge experts are key for developing shared problem domain understanding. They share their relevant knowledge and refine their own knowledge and others' understanding by a process of sharing and verifying the understanding of that knowledge and how it may affect the software product. The vendor stakeholders may also have previous domain knowledge and experience (from previous software development projects in a similar domain) that can add to or develop a clearer picture of the problem domain's needs and issues.

It is interesting to note that a number of the participating organisations in this study employed domain experts who were past or potential clients, and they often took the role of "proxy" client knowledge providers.

It is also worthy of mention that one of the strong themes to emerge from this study is the importance of the quality of the client representative(s), who is the main participant in interactions with the vendors. Part of the notion of a “low quality” client representative related to the client representative being a poor knowledge provider (i.e. having insufficient domain knowledge). This was identified as a significant barrier to efficiently developing an appropriate level of client-vendor shared understanding. This is discussed in detail in Chapter 4.

2.3.1.2. Mediator

The role of mediator is related to the concept of someone acting as a “bridge” between the majority of the vendor and client groups. The gap referred to exists because of the contrasting characteristics of the stakeholder groups which may relate to cultural attributes such as cultural background (Damian, 2007) or technical attributes, such as difference in technical or domain knowledge (Aranda, et al., 2007a; Saiedian & Dale, 2000).

This particular role is commonly seen on the stakeholders from the vendor side, such as: business or system analyst, project manager, marketing representative, consultant, or any other stakeholders who usually engage in communication with the client group (for example see Aranda et al., 2007a). According to Cao and Ramesh (2008), stakeholders with this role usually sit together and communicate directly with the client stakeholders, either on full-time or part-time basis. Later, they return to the vendor group and act as a surrogate customer or proxy client in communicating the knowledge (e.g. requirements) obtained to the rest of vendor team.

Literature (e.g. Curtis et al., 1988; Emam & Madhavji, 1995) indicate that it is important for the person given this particular role to possess good communication skills as well as to have relevant depth of experience of the business domain. Typically they also have sufficient technical knowledge to be able to share a “common” language with the developers and to ensure that they know what kind of knowledge they might need.

Moreover, to lessen the effect of cultural differences, it is of benefit to sharing understanding if the mediator has the same or similar cultural background to the people from the client group, as suggested by Damian (2007).

2.3.1.3. Facilitator

Another role identified from the literature is termed a 'facilitator'. A facilitator is someone who tries to make sure that communication and knowledge transfer occurs by providing sufficient opportunity and encouragement to share. Coughlan et al. (2003), who investigate communication issues during requirements elicitation, report that this role is usually identified with:

- Project managers who are usually in charge of arranging meetings between the vendor and client groups.
- Business consultants who are responsible for ensuring that any relevant knowledge is specified and represented appropriately, to maximise likely understanding.

Therefore the notation of the representation of knowledge and its level of abstraction are clearly significant influences on the ease and level of shared domain understanding, as is the timing and frequency of the opportunities to share and test this understanding. This is supported well in the results of this study where the mode of representation and timing of interactions were seen as possible barriers to shared understanding.

2.3.1.4. Regulator

Another role which is identified in literature such as Alexander and Robertson (2004) and Glinz and Wieringa (2007) has been labelled as 'regulator'. This particular role is usually present in a person or organization such as government agencies, standard institutions, and legal-bodies, which provide (and enforce sometimes) rules or regulations that project members have to obey or implement in the product being developed and are not negotiable. Such compliance rules represent knowledge that has to be gained by the vendor through the client or regulator directly and the implications for the software application need to be understood.

This was not identified as a significant issue in sharing understanding by the organisations participating in this study.

2.3.1.5. Decision maker

Throughout a development project decisions regarding future actions are constantly being made. This may range from feature selection and prioritisation to “go-no go” decisions. For instance, often decisions have to be made regarding which of multiple points of view on particular requirements or domain knowledge will be used, as described in Damian (2007). Sharp et al., (1999) note that the stakeholder role of decision maker, with a high level of decision making capability and the appropriate authority to make that decision, is necessary in order to keep a project running. This particular role is may be assumed by project managers, project’s sponsor or project’s financial controller.

In terms of shared understanding, the relevant knowledge to make the decision needs to be shared, and understanding of the result of the decision, often with justification, needs to be shared.

The notion of a decision making was identified as another dimension of a quality customer representative in this study. Participants identified a number of potential barriers to shared understanding related to this. A poor quality customer representative may make a decision related to refining shared understanding without the appropriate authority, resulting in either back-tracking or product errors. Similarly a customer representative who makes ill-informed or unsupported decisions was seen as negatively influencing shared understanding

2.3.2. Stakeholder-related problems in developing shared understanding

Unsurprisingly, literature identifies several issues related to the stakeholders that can be associated with the process of developing shared understanding. These are useful as potential barriers to compare with those identified by the participants of this study.

The first problem is regarding to the availability of the key client stakeholders who are able to provide high value knowledge during the project (Cao & Ramesh, 2008; Emam & Madhavji, 1995). This can be related to the client stakeholders' commitment to the project (Coughlan et al., 2003) or it can be due to the client stakeholders' uncertainty (Emam & Madhavji, 1995; Saiedian & Dale, 2000). Consequently, if there are no client stakeholders who able to contribute useful information, the vendor needs to probe for the information themselves. One technique recommended by literature is to present something concrete that might resemble the clients' needs (e.g. prototypes, similar products), which is is useful for triggering ideas from the client stakeholders (Emam & Madhavji, 1995; Nuseibeh & Easterbrook, 2000; Saiedian & Dale, 2000).

Other researchers, for instance: Curtis et al (1988); Hall, Beecham and Rainer (2002b); and Lubars et al. (1993), show that a lack of domain knowledge on the vendor side can be a problem as well. This vendor's lack of domain knowledge may affect the ability of the vendor in absorbing and interpreting the domain knowledge shared by the client group. Lubars et al. (1993) suggest that the most efficient method for handling this problem is for the vendor to employ a domain expert whose function would be to bridge the communication gap between the vendor and client groups (rather than developing the development group's domain knowledge incrementally as the project progresses (Curtis et al., 1988).

Another problem identified in literature is related to the existence of multiple points of view or mental models within the client group, which sometimes can be in conflict (Alcázar & Monzón, 2000; Curtis et al., 1988). Curtis et al. (1988) go on to argue that this problem is mainly a prioritization issue. This problem is also noted in some empirical studies such as those done by and Lubars et al. (1993); Coughlan et al. (2003); and more recently Cao and Ramesh (2008). In terms of dealing with this problem, one participant from Cao and Ramesh's study (2008) forced all the involving stakeholders to be physically present in a series of meeting in order to obtain a consensus. This particular approach seems effective if the vendor has enough authority over the client stakeholders.

Cao & Ramesh, (2008) and Damian, (2007) uncover another problem, related to the level of trust between the vendor and the client groups. Damian (2007) in her study argues that this issue of trust is strongly connected to the stakeholders' inability to be open with the other stakeholders group. She found this to be strongly influenced by cultural factors such as differing attitudes towards hierarchy and communication styles.

According to Cao and Ramesh (2008), in the context of Agile development, one way to improve trust is by producing artefacts that represent a commitment and understanding, such as appropriate documentation. Cao and Ramesh go on to point out that the overheads in keeping such documentation up-to-date with rapidly evolving understanding or high uncertainty can be counter-productive. Curtis et al., (1988) and Damian, (2007) demonstrate that specific efforts to develop the inter-group relationship (e.g. social activities, informal communication) can improve the level of inter-group trust.

Another significant factor identified in literature that influences the development of shared domain understanding is related to the actual activities the stakeholders undertake that potentially contribute to or inhibit the sharing of domain understanding between the client-vendor groups. This is the subject of study in the next section.

2.4. Development of shared understanding activities

In the context of developing shared understanding, it is clear from literature (e.g. Coughlan et al., 2003; Damian, 2007; Hsieh, 2006) that communication activities play an important role in developing shared understanding. These activities typically include knowledge acquisition and knowledge sharing activities (Coughlan et al., 2003; Damian, 2007), as well as knowledge verification and validation activities (Damian, 2007). As well as the activities themselves, a number of researchers (for example Arthur & Gröner (2005); Curtis et al. (1988); Deshpande et al. (2005); Coughlan et al. (2003); Rickman (2001); Lubars et al., (1993) and; Zowghi et al., (2001)), demonstrate that it is important that these activities occur throughout the software development process iteratively, with suitable frequencies and timeliness. The general argument supporting this recognises that stakeholders' understanding of the application domain is shaped and refined throughout the phases of the software development lifecycle and this requires regular communication and "testing" of the understanding between each group.

In cases of geographically distributed software development projects these communication activities utilise indirect modes such as e-mail, instant messaging, or phone (Aranda et al., 2007a; Damian, 2007; Zowghi et al., 2001), which these researchers generally agree are less rich than face-to-face meetings. However, with the emergence of internet communication technology such as the teleconferencing, this problem can be alleviated, even though it is not as good as face-to-face communications (Damian, 2007; Zowghi et al., 2001).

It is often argued that the process of developing shared understanding is better to be done face-to-face (Cao & Ramesh, 2008; Coughlan et al., 2003; Curtis et al., 1988; Lubars et al.'s, 1993). Coughlan et al. (2003) argue that face-to-face communication may reduce the level of ambiguity as well as providing the opportunity to detect some non-verbal cues, such as: stakeholders' body language, which may influence the quality of the communication process. Similarly, Cao and Ramesh (2008) also report that face-to-face communication, in the context of Agile software development, is needed to monitor the client's evolving understanding and subsequent requirements. They provide evidence that this face-to-face process can replace the documentation of this evolving understanding and requirements and can be more efficient.

Since this research is investigating the issues around the development of shared application domain understanding between client and vendor groups in software development, particularly during the Requirements Engineering process, activities related to sharing understanding of the solution domain are largely left unexplored. Activities related to knowledge creation, sharing, acquisition and confirmation about the problem domain are the focus, as shown in figure 2.4.

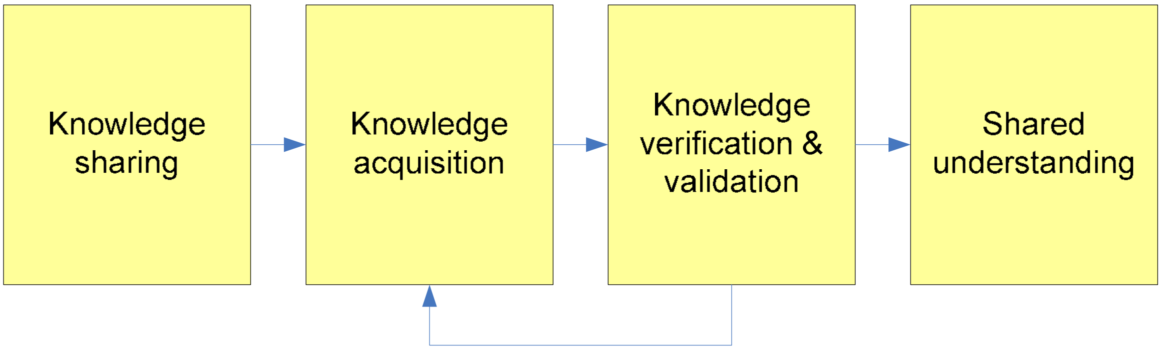

Figure 2.4 – Development of shared understanding activities

These activities can be viewed from a number of theoretical perspectives, for example communications theory or knowledge management theory. In the subsequent sections the activities have been grouped into those related to knowledge sharing and acquisitions, and those related to knowledge verification for analysis and discussion. Tools and techniques described in literature, related to these distinctions, are also summarised.

2.4.1. Knowledge sharing and acquisition

One of the important activities in order to develop shared understanding between two persons or groups is knowledge sharing and acquisition. In terms of developing shared understanding between the vendor and the client groups, the process of knowledge sharing and acquisition typically starts with what Arthur and Gröner (2005) call the indoctrination phase, where the representatives from the client group share an overview of their problems and needs. During this activity, the representatives of the vendor group are "acquiring" the knowledge being shared using the new knowledge to test and refine their world views and "mental maps". This may uncover areas of ignorance or uncertainty related to the problem domain that trigger the vendor stakeholders to elicit more knowledge from the client stakeholders and share this. In summary, these activities of knowledge sharing and acquisition allow the exploration of the clients' needs or goals, the problem domain and perhaps possible solution approaches that the clients have in their minds (Damian, 2007).

In light of the requirements engineering process, knowledge sharing and acquisition activities mainly occur during the elicitation phase, since elicitation activities focus on capturing the knowledge from key stakeholders (Nuseibeh & Easterbrook, 2000). In eliciting the knowledge from the clients, various techniques are employed. Nuseibeh and Easterbrook (2000) developed an often cited classification of elicitation techniques which is used to structure the examples provided in Table 2.2. This is combined with information from Zowghi & Coulin's study (2003) regarding the suitability of the techniques from each category for understanding the problem domain.

No.	Technique category	Description	Suitable for domain understanding
1.	Traditional techniques	Includes questionnaires, surveys, interviews, and analysis of existing documentation.	Yes
2.	Group elicitation techniques	Includes brain storming, focus groups, and consensus-building workshops (RAD/JAD workshop).	Yes
3.	Prototyping	Includes working prototype, screen mock-up, web based prototype.	No
4.	Model-driven techniques	Uses a model to drive the elicitation techniques, includes goal-based and scenario based methods.	Yes
5.	Cognitive techniques	Includes protocol analysis, laddering, card sorting, and repertory grids.	Yes
6.	Contextual techniques	Includes participant observation, ethnography, and conversation analysis.	Yes

Table 2.2 - Classification of elicitation techniques and their suitability for domain understanding (adapted from Nuseibeh & Easterbrook, 2000; and Zowghi & Coulin, 2003)

Interestingly, Zowghi & Coulin argue that prototyping-related techniques are not suitable for improving domain understanding, yet this was cited as a common technique to improve domain understanding by the participants of this study. This is discussed more fully in Chapter 4.

A. Possible tools and techniques

Based on existing field studies, (see for instance: Aranda et al. (2007a); Cao and Ramesh (2008); Coughlan et al. (2003); Hofmann and Lehner (2001); Neill and Laplante (2003)), traditional techniques are found to be very common in knowledge acquisition activities For example, the majority of the field studies report that the main technique employed by the vendor for obtaining knowledge from the client group is an unstructured interview. According to Coughlan et al. (2003), interview is popular due to its ability to obtain the knowledge in detail. In some of the practice reported, the interview often resulted in the development of user stories (Aranda et al., 2007a; Cao & Ramesh, 2008).

In addition to traditional techniques, literature also indicates the popularity of group elicitation techniques such as focus group and consensus-building workshops (e.g. Joint Application Design (JAD) workshop, User Centred design workshops) in practice. For instance see: Hofmann and Lehner (2001); Lubars et al. (1993); Neill and Laplante (2003)). Typically, the employment of group elicitation technique is aimed to support the development of agreement or buy-in among the participating stakeholders. However, it also opens the opportunity to exploring wider range of ideas from the participants, thus triggering richer understanding of participants needs (Nuseibeh & Easterbrook, 2000). This ability to exploring participants' ideas is particularly suitable for product or market driven organisations which typically do not have any fixed customers, as is reflected in Lubars et al.'s study (1993). According to Coughlan et al.'s finding (2003), however, group elicitation techniques, particularly workshops, are ineffective for acquiring details of knowledge. They show that group discussions at a very detailed level generally lead to more conflict or disagreement, rather than buy-in and agreement. Consequently, they argue, group elicitation techniques are most useful for eliciting and discussing the "big picture" of the problem domain, staying at at higher levels of abstraction.

The use of prototype and model based techniques as described in table 2.2 are found to be more common during the knowledge verification and validation phases of the process (e.g. see Hofmann & Lehner (2001)). A number of researchers, however, relate prototyping to knowledge elicitation, particularly if there is a high level of uncertainty coming from the client side (Berzins, Shing, Riehle, & Noguiera, 2000; Emam & Madhavji, 1995; Nuseibeh & Easterbrook, 2000). They argue that working with a prototype may trigger the elicitation of further requirements or clarification of domain knowledge.

There seems to be little evidences identified from the literature regarding the use of cognitive and contextual techniques in practice, even though researchers recommend the use of those techniques.

Literature also indicates some tools that can support knowledge sharing activities among the stakeholders. One set of tools identified from the literature are software applications such as Rational Requisite Pro, Telelogic Doors and other RE management tools. This kind of tool supports interactive collaboration and communication between stakeholders during the project life cycle by providing features such as an interface for communication, a knowledge or requirements repository, as well as other RE related functions such as prioritization, change management. Damian (2007) presents evidence that this kind of tool is effective for facilitating interaction between stakeholders in a distributed development project. However, according to Hofmann and Lehner's findings (2001), this kind of tool can actually disturb the RE processes, by adding to the complexity and cognitive effort resulting in confusion to its users, particularly if there is lack of well-defined RE processes and lack of the training of the tools.

Other common tools identified from the literature are whiteboards (Aranda, et al., 2007a), internal web-sites (Aranda, et al., 2007a; Hofmann & Lehner, 2001) and Wikis (Aranda et al., 2007a; Decker, et al., 2007). All of these tools are useful as knowledge repositories, which can be accessed by the project members. To some extent, these kinds of tools are also useful for informing all the project members about the progress of the project (Aranda et al., 2007a; Decker, et al., 2007).

2.4.2. Knowledge verification and validation

Other activities which are significant for developeing shared understanding are knowledge verification and validation. Knowledge verification and validation involves confirming the obtained knowledge with the relevant stakeholders and letting the stakeholders validate the vendor group's understanding. This improves the consistency and accuracy of the knowledge obtained from the client group by (Nuseibeh & Easterbrook, 2000). The vendor group's understanding can be represented in various forms, including documents, prototypes. Further investigation of the representation of understanding is addressed in a later section of this literature review. In the situation where the project required the vendor to engage in communication with more than one client group's stakeholders, Hammond et al. (2001) stressed that, it is important to contact all of the relevant stakeholders for confirmation purpose, in order to ensure the accuracy of vendor's understanding on the knowledge gathered.

Much of the literature indicates that the process of verification and validation of the knowledge should be iterative (for instance see: Damian (2007); Rickman (2001)). This is very important as the iterative process allows the stakeholders, particularly the stakeholders from the vendor group, to reshape their understanding of the knowledge obtained in the previous occasions (Damian, 2007), as well as enhancing their current knowledge due the discoveries of deeper knowledge which is triggered by the validation processes. During the iterations of verification and validation activities, it is common that the process would include negotiation activities (Coughlan et al., 2003). These activities focus on establishing a consistent view of the shared knowledge by having stakeholders compromise and concede their views, especially if there are many client stakeholders with different point of views (knowledge) involved in the process of knowledge sharing.

A. Possible tools and techniques

As is indicated by a number of researchers (e.g. see Hofmann & Lehner, 2001; Saiedian & Dale, 2000), verification and validation processes are usually accompanied by the employment of artefacts that can support the communication between the client and the vendor groups. The most common artefacts identified in the literature are prototypes (e.g. see Hofmann & Lehner, 2001) and documents (e.g. see Aranda et al., 2007a; Curtis et al., 1988, Hofmann & Lehner, 2001). In some cases, the product in development itself can also be the artefact (e.g. see Cao & Ramesh, 2008).

A prototype, which can range from screen mock-ups to working prototypes, is one of the most commonly used artefacts used during the process of verification and validation and is the subject of a considerable amount of research attention, for example Aranda, et al. (2007a); Cao and Ramesh (2008); Emam and Madhavji (1995); Hofmann and Lehner (2001); Lubars, et al. (1993); and Neill and Laplante (2003). According to Hsia, Davis and

Kung (1993); from Saiedian & Dale, 2000), it is believed that the use of a prototype could enhance the understanding of a problem, as well as identifying appropriate and feasible external behaviour that might be useful for determining possible solutions.

These particular benefits of using prototype are most likely gained by complementing the use of prototype with a walkthrough or a review technique (Hofmann & Lehner, 2001). This is a technique for looking through the understanding embodied within an artefact, to gauge the "synchronization level" between the vendor and client groups' current state of understanding. Since the process of reviewing the artefacts usually triggers some discussion, this will generally enhance the stakeholders' current state of understanding of the problem domain.

Similar to a prototype, a walkthrough is a technique that is identified in literature (e.g. Aranda et al., 2007a; Cao & Ramesh, 2008; Curtis et al., 1988; Hofmann & Lehner, 2001; Neill & Laplante, 2003). Moreover, according to Hofmann and Lehner's findings (2001), walkthroughs are usually done by successful RE teams and are recommended as best practice.

Activities related to the verification and validation of shared understanding can also be supported by the use of:

- Scenarios (Neil & Laplante, 2003; Hofmann & Lehner, 2001), which are useful for examining the domain knowledge (particularly the requirements) in context.
- User acceptance tests for validating the knowledge (particularly the requirements), which usually participated by the real users or QA personnel (Cao & Ramesh, 2008).

2.4.3. Other activities that support the development of shared understanding

Beside the two main activities mentioned in the last sections, literature also indicates that some other activities that able to provide support in the development of shared understanding.

The first activity that is considered useful for developing shared understanding is informal communication, which refers to any other communication activities outside the formal communication activities such as scheduled meetings. Curtis et al., (1988) demonstrate that informal communication, which can include activities such as a phone call or casual conversation during lunch time, is effective for transferring knowledge across organization boundaries. Moreover, Damian, (2007) shows that informal communication is also useful for building relationship between the stakeholder groups. This relationship development, if maintained regularly can eventually result in higher trust level from the client stakeholders (Cao & Ramesh, 2008). This can potentially enhance their willingness to be involved in the project and encourage them to provide a higher level of detailed domain knowledge. Furthermore, Damian (2007) points out that informal communication is advantageous for probing for information, particularly when there are political or personal agendas behind the requested project.

The second activity identified that supports the development of shared understanding is related to proper planning and preparation. During the planning and the preparation it is recommended by Damian (2007) to identify all of the relevant stakeholders and their roles and responsibilities and share this with other project members. This assists the maintenance of suitable communication channels by allowing the easy identification ofw ho to contact if there is a need for eliciting more knowledge, confirmation, or if there are problems.

Other activities that are recommended in literature include the preparation of a vocabulary list or glossary and the use of templates. The preparation of vocabulary list or glossary is useful particularly if there are is a lot of specialised vocabulary (jargon) associated with the problem domain (Damian, 2007; Offen, 2002). Preparing a vocabulary in the early phase is recommended to provide an accessible reference point for involving stakeholders to avoid confusion during the project. While, preparation of templates is recommended to provide a common mean for communicating knowledge that can be used by stakeholders both from the client and vendor groups (Damian, 2007).

2.4.4. Problems in development shared understanding practice

In terms of how practitioners develop shared understanding, literature identifies some of the barriers that stakeholders might face related to the practices and activities.

Some researchers (e.g. Damian, 2007; Zowghi et al., 2001) address the issues of large geographical distances and time zone differences between the stakeholders. This situation, with low or no face-to-face communication, can affect the quality of the communication between the two separated groups. Typically this is addressed either by bringing the groups physically together, which can be very costly, or use of a video conferencing facility, which still loses some of the richness of face-to-face contact.

This was identified as an issue in the current study and the preference was definitely for face-to-face meetings, although internet based video conferencing was also used by some participants.

Proxy clients or surrogate customers for bridging the communication between the client and the vendor groups is another common practice identified in literature (e.g. see Aranda et al., 2007a; Cao & Ramesh, 2008; Curtis et al., 1988). According to some of the studies (e.g. Curtis et al., 1988; Damian, 2007; Lubars et al., 1993), due to the high reliance on the proxy client’s interpretation, there is a high chance of misinterpretation being propagated through the vendor group. The only approach recommended from the literature regarding this problem is by employing a domain expert for supporting the interpretation process (Curtis et al., 1988; Lubars et al., 1993). According to Curtis et al. (1988), besides possessing good knowledge of the domain, it is also essential for the domain expert for having good communication skills and willingness to engage in face-to-face communication with the client side.

Another common practice addressed by the literature is the employment of prototype in the RE process. This particular practice is believed can support the elimination of ambiguities and inconsistencies (Hofmann & Lehner, 2001), which to some extent can support the development of shared understanding. However, existing literature also identifies some limitations or dangers related to the use of prototype. The first limitation concerns the limited knowledge that can be expressed by a prototype. For example: Cao and Ramesh (2008) found that the use of prototyping often neglects some of the non-functional requirements such as its scalability, security, and robustness, that might be relevant to the problem domain. Secondly, in utilizing a prototype, there is a higher risk that it can trigger excessive customers' ideas, which could lead to unrealistic expectations from the client groups (Cao & Ramesh, 2008; Lubars et al., 1993). This could hamper the development of shared understanding due to over-frequent changes to understanding and client groups' needs. Thirdly, as previously mentioned, Hofmann and Lehner (2001) argue that the use of a prototype can help in eliminating ambiguities and inconsistencies; however this particular strength could be a double edged sword. The existence of a prototype could limit the customer's view of other ways of understanding a particular problem, because of high influence of the particular understanding within the prototype. In addition, Emam and Madhavji (1995) argue that the use of a prototype can be an addition to the project's costs, which could prove to be a problem for a project with a tight-budget.

Lastly, another barrier identified in the literature is related to the difficulty to access the primary source of the knowledge (Cao & Ramesh, 2008). One of the possible causes of this problem is the existence of multiple layers of stakeholders. For example the development team may have to deal with a subcontractor (or other third party) to the client stakeholders (Lubars et al., 1993). Lubars et al. go on to suggest that a training session with the client can be used as an opportunity to ensure direct contact with the client.

2.5. Representation of understanding

According to Curtis et al. (1988), a successful project usually employs a common representation of knowledge, which is useful for facilitating communication and providing common references for discussions. There is certainly considerable research in the area of representing knowledge for sharing and improving understanding. It is clearly identified in literature as an important factor in effective development of shared understanding and is closely related to the notion of having a common "language" to share, verify, test, manipulate and discuss understanding.

From literature, the representations of current understanding may range from informal representations, which are loosely structured and have few constraints on their notation (such as natural language), to semi-formal representations (e.g. use cases) and formal representations with a high degree of structure and constraint (such as object-oriented models). A recent survey of RE practices by Neill and Laplante (2003) indicates that informal representation types still dominate practitioners' choices compared to formal ones. Moreover, they also found that the use of informal representation seems to support the end-product's quality better in terms of usability and the fulfilment of customers' needs. This finding seems to be in contrast with another study by Hofmann and Lehner (2001) which indicate that successful RE team appeared to employ advanced modelling (e.g. Object Oriented models), which tend to have high degree of formality. However it is also reported that the same successful RE team also employed basic models (e.g. entity-relationship models/diagrams and state transition diagrams) and prototypes as the representations of knowledge, which are considered more effective in supporting the communication between the vendor and client groups. This indicates that the employment of representation with high degree of formality is not a significant factor that determine the success of a RE team, and perhaps it is better for practitioners to employ more than one type of representation.

In the next sub-sections, types of representations are identified from literature and their significance to shared problem domain understanding is investigated in more detail. These have been categorised into natural language, graphical, and product-related representations for the purposes of this discussion.

2.5.1. Natural language representations

Natural language representations are the most common representation used by practitioners, either as verbal communication (e.g. conversation) or text (e.g. documentation), see for instance : Aranda et al (2007a), Hammond et al. (2001), Hofmann and Lehner (2001), Karlsson et al. (2007). The specific format of the written natural language representations of shared understanding is wide ranging, including user stories, scenarios, requirements specification documents, business process specifications, operations manuals, strategic plans, ‘request for change forms’, emails, online discussions and so on.

According to researchers (e.g. Rickman, 2001), the use of natural language is very important for describing the knowledge that needs to be interpreted. It is a natural mode of communications, as writing and reading, and has a very familiar notation and set of rules. Furthermore a very wide diversity of ideas and concepts can be expressed using natural language. Gervasi and Sawyer (2008) share this view maintaining that it can be an effective representation for sharing understanding because the properties of the representation are generally shared by all of the stakeholders.

One of the problems discussed in literature concerns the misinterpretation or ambiguity of natural language representations. According to Karlsson et al.’s (2007) findings, this particular problem is strongly related to the “richness” of natural language. They go on to suggest an approach to alleviate this problem is to have constant or regular clarification activities, such as review meetings or group discussions. In addition, the employment of a glossary or vocabulary can be useful for providing references to the “local understanding” of particular words or terms (Alcázar & Monzón, 2000; Offen, 2002).

2.5.2. Graphical representations

Another type of representation identified in the literature is graphical representations. This kind of representation is often found in form of models, and is common in RE activities (Nuseibeh & Easterbrook, 2000). A model is considered important for understanding the (application) domain of a project, as it may support the enhancement of practitioners’ understanding by providing references or representation on how things interact with the system in the real world or application domain (Hammond et al., 2001; Offen, 2002; Rickman, 2001).

Based on the literature reviewed, many types of models related to requirements engineering are identified, including: context diagrams (Hammond et al., 2001; Rickman, 2001), Entity Relationship (ER) diagrams (Alcázar & Monzón, 2000; Davies et al., 2006; Hofmann & Lehner, 2001; Lubars et al., 1993), data flow diagrams, flowcharts, workflow diagrams (Davies et al., 2006), and various object oriented models (Cederling et al., 2000; Davies et al., 2006; Hammond et al., 2001; Lubars, et al., 1993). Each of the types usually intended to serve a particular purpose. For instance: ER diagrams are intended to depict the relationship between entities related to a system, workflow diagrams are intended to depict work flow of a particular system or process, and so on.

In terms of representing the domain, the most common graphical representation used is probably context diagrams (Hammond et al., 2001; Rickman, 2001). According to Hammond et al.'s experience (2001), context diagrams easy to understand by the client stakeholders, since the notation is simple (has few elements and rules) using only a combination of at boxes, arrows and text. However, they also argue that context diagrams are very constrained in terms of only allowing practitioners to show entities that directly connect with the system. Therefore, for the sake of completeness and reducing ambiguity they recommended creating the context diagram for all of the entities involved in the domain, as it is found to be effective in avoiding misunderstanding during the knowledge confirmation process.

Another common graphical representation found in the literature is object oriented (OO) modelling, which is typically in the form of UML notation (e.g. class diagrams, state diagrams, and sequence diagrams) (Cederling et al., 2000; Davies et al., 2006; Hammond et al., 2001; Lubars, et al., 1993).For example, a state transition diagram (Lubars, et al., 1993) and a sequence diagram (Hammond et al., 2001) are useful for describing the dynamic behaviour of the application domain, such as its control flow, while a class diagram is more suitable for representing more static information, such as domain properties and interfaces between entities (Hammond, et al., 2001). In terms of its usage, some of the literature reviewed (e.g. Alcázar and Monzón (2000); Offen (2002)) indicate that the use of OO modelling is effective for understanding the problem domain due to its lack of ambiguity compared to natural language representations. In addition, Rickman (2001), who also recommended the utilization of OO modelling, added that OO model is able to provide more insights on the domain, such as functionality that needs to be performed on the data, so that more understanding is gained by the stakeholders.

However, concerning the utilization of OO model, which typically represented in form of UML notation, it is interesting to notice that from a survey conducted by Davies et al. (2006), more than half of the participants (from 312 complete responses) did not know or did not use UML. Unfortunately, the study did not investigate the reasons. Another negative reception toward UML is also identified in Karlsson et al.'s (2007) study, which found that client stakeholders often lack the skills to interpret, manipulate and test such representations. These stakeholders typically are clients with no or little software development background.

Lubars et al. (1993) show that difficult or unfamiliar representation can cause difficulty for the stakeholders in accessing information from the representation being used. This is closely related to the stakeholders' experience or skill in dealing with particular types of representation and is often associated with the use of advance modelling method, such as: UML diagrams, object oriented models, etc. (Emam & Madhavji, 1995). One approach for alleviating this problem is described by Lubars et al. (1993) where they suggest developing the stakeholders' skills in dealing with this kind of difficult representation through training. This could add considerable time and resource costs toa software development project, however. Therefore, using another type of representation, which is less complex and easier to comprehend (e.g. prototype, natural language, etc.) might be more feasible, as suggested by Emam and Madhavji (1995).

Overall the evidence points to OO modelling being more effective for use within the vendor environment for analysis and communication, rather than for sharing and confirming understanding with the client group.

2.5.3. Product-type representations

According to Deshpande et al. (2005), representations of understanding which can actively be manipulated by the stakeholders (i.e. are interactive) support the development of shared understanding better than a non-interactive representation. In the context of software development project, this representation and interactivity is often through the demonstration of prototypes which resemble a possible solution product or existing similar products.

The utilizations of prototypes or executable products for visualizing the development team's current understanding appear to be very common in practice (For instance see: Aranda, et al., 2007a; Cao & Ramesh, 2008; Cederling, et al., 2000; Emam & Madhavji, 1995; Hofmann & Lehner, 2001; Lubars, et al., 1993; Neill & Laplante, 2003). This most likely because this type of presentation is very concrete and testable thus supporting the generation of feedback among client stakeholders, as noted by some researchers (e.g. Emam & Madhavji, 1993; Karlsson et al., 2007).

2.6. Conclusion

A review of current related literature confirms that shared understanding of the problem domain between the vendor and the client group is important for ensuring the success of a project and is still problematic. However, there seems to be limited literature that focuses on the practice of developing shared understanding, particularly in the context of software development. This, therefore, supports the contention of this study to investigate this area.

An analysis of the literature reveals that there are several factors that can significantly impact on the quality and depth of the development of shared understanding. These ideas have been synthesised in figure 2.5, which represents a model of these factors. Although the dimensions shown in Figure 2.5 may overlap and influence each other, it provides a useful framework that provides some insights to the comprehension of current practice in sharing understanding. This framework will form the basis of the design of the field study and a comparison of the results with literature. Before presenting the results and discussing them, however, the research methodology and techniques used in this study are described and justified in the next chapter.

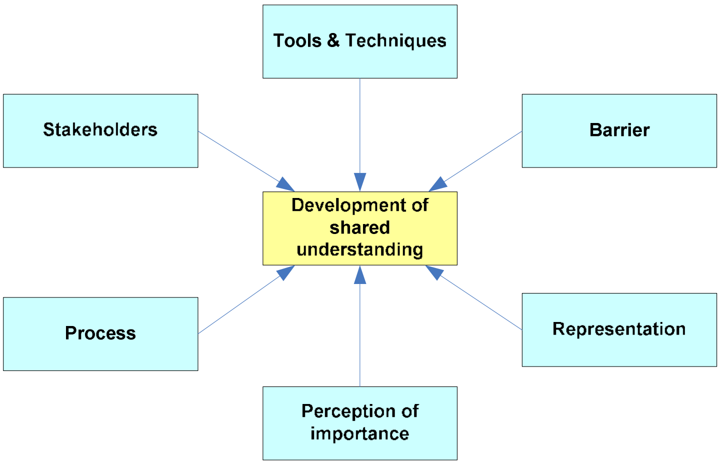

Figure 2.5 – Dimensions that are relevant to development of shared understanding

Chapter 3

Research Methodology

3.1. Introduction

The selection of a research methodology should be aligned with the philosophical and theoretical frameworks within which the phenomenon under investigation is operating and the nature of the research questions about this phenomenon. The selection of particular research methodology strongly influences the selection of research methods used to conduct the research including the data collection and analysis methods.

This chapter positions this research in terms of research methodology and justifies the suitability of the selected research methodology. Several candidate methods are evaluated and a multiple case study approach is justified in section 3.3. Rationalisations of the research design and research questions are presented in section 3.4, including the design of a semi-structured interview and participant selection criteria. A description of the protocol for conducting the field work and collecting the data is provided in section 3.5 and how this relates to the research design is discussed. The methods of analysing the collected data is explained and defended in section 3.6. The chapter concludes with a summary of the main research decisions and their justifications and introduces the Chapter 4, the results and their analysis.

3.2. Research type and paradigms

It is the intention of this research to gain some insights into the phenomenon of shared understanding development through practitioners' perceptions. The specific context of this study relates to the evolution of problem domain understanding between a client stakeholder group and a vendor group who are developing a software application (or product) to meet the client's needs. This is expected to provide insights that will be useful for improving current software development practice and may suggest potential fruitful areas of future research.

A number of research approaches could be taken to achieve this aim and answer the research questions described in Chapter 1. The approach taken in this study is to understand the phenomena through people's complex perceptions in giving meanings to the process of shared understanding development (including: activities, artefacts, etc.). This focus on practitioners' perceptions rather than for example direct observation or document analysis, is based on a desire to "know" the practitioners more closely (as "customers" of RE research) and understand their experiences and needs in this area. For this type of study involving the investigation of people's interaction within their social and cultural context the adoption of an interpretivist epistemology is common and is well accepted in Information Systems research (see for example Klein & Myers, 1999; Aranda et al., 2007b; Coughlan et al., 2003). This study therefore adopts an interpretive research approach.

It is common to distinguished research approaches as either qualitative or quantitative where qualitative research has its focus on social and cultural phenomena and deals with collecting and analysing qualitative data and quantitative research focuses on natural phenomena and deals more with quantitative data (Myers, 1997). In terms of this study a qualitative research approach is taken, the data to be gathered and analysed is the vendors' perceptions of their experiences of developing shared understanding with the client group.

Within this research paradigm there are a number of candidate methods and techniques for gathering and analysing the data. These are discussed in the next section and the use of a multiple case study approach involving semi-structured interviews is justified.

3.3. Research method

This study aims to study practitioners of software development involved in commercial projects and so an interpretive field study (Klein & Myers, 1999) was chosen to be the research method, since it offers the opportunity for studying the phenomenon in its commercial context in the real world.

A field study usually conducted in form of case study (e.g. see Curtis et al., 1988; Walsham, 1993 as cited from Klein & Myers, 1999) or ethnography (e.g. see Suchman, 1997; Wynn, 1979; Wynn, 1991; Zubot, 1988 as cited from Klein & Myers, 1999). In term of this research, case study is preferred because of its flexibility for not depending solely on detailed observation and also it does not require spending significant periods of time on the field, as is required by ethnography research (Klein & Myers, 1999). This is certainly suitable considering the given time limit allocated for conducting this research.

According to Yin (1994 as cited from Tellis, 1997b), a case study can be used for several purposes, namely:

- To explain complex causal links in real-life interventions
- To describe the real-life context in which the intervention has occurred
- To describe the intervention, or
- To explore those situations in which the intervention being evaluated has no clear set of outcomes.

In other words, a case study can be used as explanatory, descriptive or exploratory research. As for this research, considering the research's aim for uncovering the phenomena and also because there seems to be little information on the phenomena itself (Collis & Hussey, 2003; Tsai & Gururajan, 2007), especially in the context of software development project, it appears that the term exploratory is suitable for this research. In fact, it has been well-acknowledged that case study is a popular method for conducting exploratory research (Collis & Hussey, 2003; Leedy & Ormrud, 2005 as cited in Tsai & Gururajan, 2007; Voss et al., 2002).

Moreover, in term of epistemology, a case study is found to be suitable for both positivist (e.g see Yin, 2002 as cited from Myers, 1997) and interpretive epistemology (e.g. see Walsham, 1993 as cited from Myers, 1997).

This alignment with purpose and epistemology provide strong motivation for selecting a case study as the research method for this study.

In the practice, a case study research can de designed to deal with a single case only or multiple cases. By focusing on single case, a researcher is generally able to obtain deeper understanding of the phenomenon under investigation compared to focusing on multiple cases. On the other hand, focusing on multiple cases opens the opportunities for researcher to identify various points of views (Herbert, 2005, p.41). This could increase the possibility of discovering new variables related to the phenomenon, thus increase the possibility to discover future research opportunities. Moreover, focusing on multiple cases also increases the chance of obtaining what Stake (1995, as cited from Tellis, 1997b) defined as “naturalistic” generalization. This type of generalization is based on harmonious relationship between the readers’ experiences with the cases studies represented. Therefore, in order to maximize the contributions both to researchers and practitioners, the research is decided to focus on multiple cases.

In conducting this multiple-case study research, each case study is planned to be done following an established protocol, which covers field procedures, data to be collected, data collection method, research instruments, and so on. Discussion of the protocol will be covered in the following section. This kind of replication is consistent with Yin’s (2003) suggestion, however for slightly different purposes. As Yin’s (2003) guidelines of replication is more focused on theory or rival-theory testing, this study’s replication is intended to ensure the retrieval of similar set of data for comparison and pattern identification purposes during the analysis phase.

With the selection of case study as research method, the following discussion will follow Yin’s (2003) recommendations on case study stages, which start from the research design phase, conducting the case studies, and finally analysis of the data from the case studies.

Discussion of the unit of analysis as “the process of developing shared understanding” is also presented.

3.4. Research design and preparation of data collection

3.4.1. Case study protocol

In designing a case study research, Yin (2003) suggests the development of a case study protocol, especially for multiple-case study research, as it is useful for enhancing the reliability of the research and the integrity of the data, as it provides a guideline for conducting each of the case studies in the same manner. Yin’s suggested framework fo a case study protocol has been followed, including:

- **An overview of the case study research**, which outlines the research objectives, case study issues, and presentations about the phenomenon of interest
- **Field procedures**, which provides a reminder of the procedures, credentials for accessing the data source, as well as location of those sources
- **Case study questions,** which outlines the questions that researchers need to remember during the data collection phase
- **A guide for the case study report**, which focus on the outline and the format for the report.

In term of this study, the case study protocol is reflected mostly from the research instrument of this research, which is an interview. Apart from the interview questions, which represent the case study questions, the research instrument also comprises of field procedures (e.g. self introduction, responses template for helping researcher allocating the answer), as well as overview of the research (e.g. research background, goals, current understanding of the case study issue) for ensuring the interviewees understand the research aims and context. Credentials for accessing the data sources were obtained from the interviewees by getting them to read a participant’s information sheet and signing a consent form. A guide for case study report was not provided in detail, since as Tellis (1997b) suggested, it was still far from being planned during the research design phase.

The research instrument of this research can be seen in Appendix A of this thesis, as well as the credentials, which are attached in the Appendix section in form of signed consent participant forms.

3.4.2. Interviews as sources of evidence

Case study research typically involves gathering data from several sources, as shown in table 3.1, along with their strengths and weaknesses as suggested by Yin (2003). Yin also notes that the employment of multiple sources of evidence in case study research is intended for triangulation purpose which is useful for enhancing the conviction of the findings. Tellis (1997b) also argues that multiple sources of evidence in a case study improve the reliability of the data and the data gathering process (Tellis, 1997b).

Source of Evidence	Strengths	Weaknesses
Documentation	• stable - repeated review • unobtrusive - exist prior to case study • exact - names etc. • broad coverage - extended time span	• retrievability - difficult • biased selectivity • reporting bias - reflects author bias • access - may be blocked
Archival Records	• Same as above • precise and quantitative	• Same as above • privacy might inhibit access
Interviews	• targeted - focuses on case study topic • insightful - provides perceived causal inferences	• bias due to poor questions • response bias • incomplete recollection • reflexivity - interviewee expresses what interviewer wants to hear
Direct Observation	• reality - covers events in real time • contextual - covers event context	• time-consuming • selectivity - might miss facts • reflexivity - observer's presence might cause change • cost - observers need time
Participant Observation	• Same as above • insightful into interpersonal behavior	• Same as above • bias due to investigator's actions
Physical Artifacts	• insightful into cultural features • insightful into technical operations	• selectivity • availability

Table 3.1 – Type of evidences on case study research (Yin, 2003)

For this research, however, is has been decided to utilize only interviews in multiple cases as the source of qualitative data to be analysed. It is reasoned that interviews provide the richest and least uncertain source of data for a restricted time frame. Knowing what activities to observe or what documents to analyse (to get insights into the development of shared understanding) is less certain because there is little research in this area. Interviews could even suggest some possible documents and activities to observe.

However, it does not mean that triangulation is considered unimportant, and the utilization of other data sources (e.g. client stakeholders) for triangulation has been planned as a future extension of this study.

Focusing on interviews as a data collection method has the advantages of focusing directly on the phenomenon of interest by asking both open and targeted (probing) questions related directly to the phenomenon. Furthermore, utilizing interviews also opens the opportunity for researcher to enhance the validity of the result obtained by utilizing participants' quotes in supporting conclusions (Adler & Adler, 1998).

However, interviews also introduce elements of added risks that could affect the data gathered. According to Yin (2003) as shown in table 3.1, they are:

- **Participant bias –** Yin (2003) notes that such bias is common when people are the data sources. He goes on to say that this risk could happen due to inadequate construction of the interview questions, for example questions which are intimidating or unfriendly. As a result, the interviewees may become defensive or hesitant in providing information. ***Countermeasure* -** In this research, participant bias is mitigated by conducting a validation of research questions by senior researchers from RE field, as well as conducting a pilot study with one of the participants to ensure that the questions are reasonably constructed and interviewee does not have problem with them.
- **Incomplete recollection –** This risk refers to a situation where inaccurate information collected due to interviewee's poor memory. ***Countermeasure* -** In order to mitigate this risk, the interview questions were designed to enquire in a systematic way by exploring interviewee's latest experiences first to stimulate practitioners' memories, followed by questions on their overall experiences.
- **Reflexivity –** This risk refers to interviewees' tendency to give the interviewer what they (the interviewees) thinks the interviewer wants to hear. ***Countermeasure* -** This potential risk is mitigated by employing open ended

questions, particularly at the start of interview topic sections, to avoid providing hints of expected answers to the interviewees. This may be followed by further probing, but it has been identified that the answer to the probing was not “front of mind”.

Selecting interview as data collection method leads to the options for choosing between structured, semi-structured, and unstructured interview. Employing structured interview, the questions given to the interviewee would be limited to closed questions only. Consequently, it would hinder the interviewees in giving certain information that might useful for the research unless the researcher aware of all of the questions needed to be asked about the phenomenon, which is not the case for this study. On the other hand, employing unstructured interviews could limit the opportunity to focus on certain aspects of the phenomenon identified from literature and the framework of analysis. Furthermore, considering the researcher’s limited skill and experience, there would be higher possibility that the interview may goes out of context, thus consequently would hinder the researcher from understanding the phenomenon. Therefore, in term of this study, semi-structured interview appears to be the most suitable one, as semi-structured interview allows the interviewees to have their freedom in expressing and explaining their perceptions by asking open-ended questions, and also allowing the interviewers to maintain the focus of the interview by focusing on the set of questions prepared. Some researchers (Collis & Hussey, 2003, p. 170; Karlsson et al., 2007) argue that semi-structured interview allows the interviewer to explore the phenomenon of interest in a deeper manner by asking additional or follow-up questions. Therefore, considering the need to obtain deep and rich information about the phenomenon, as well as the researcher’s level of experience, the selection of semi-structured interviews as a data gathering method seems justified.

3.4.3. Design of research instrument

Following the selection of semi-structured interviews, the research instrument, namely the interview questionnaire, was developed. The development of the research instrument is based on the research questions described in chapter one and the framework of analysis developed in chapter 2 (figure 2.5).

The analysis framework identifies significant factors related to the phenomenon of developing of shared understanding (namely: stakeholders, process, representation, barrier, tools, and techniques). This has influenced the scope of the study, the structure of the designed interview, the analysis the data, as well as the report structure.

The interview questions and structure were designed through series of brain storming and discussions with the project supervisor, as well as consultation with and validation by an expert from the RE area. The final version of the questionnaire can bee seen in Appendix A.

Table 3.2 maps the interview questions to each dimension of the framework of analysis and table 3.3 shows how the interview questions relate to the research questions developed in Chapter 1.

Dimension	Interview questions
Barrier	18-24, 25, 25b, 26,30-31, 35-36, 37a
Stakeholder	6-7, 38
Process	8-18, 20-24, 25c, 26-28, 33-38
Tools & Techniques	9-15, 20-22, 23a-b, 25c, 26-32, 38
Representation	29-32, 38

Table 3.2 – Analysis framework's dimensions and interview questions relationship

Research questions	Interview questions
Do stakeholders perceive that developing shared understanding problematic (or difficult)? If so, what do they say the barriers are?	18-19, 23-25, 25b, 26, 30-31, 36-38
What do stakeholders do (processes, techniques, tools, representations) in order to develop shared understanding of the problem domain with their client?	6-18, 20-22, 23a-b, 25a-c, 26-36, 37a-c, 38
How does what the stakeholders say they do compare to what is reported in research literature?	6-38

Table 3.3 – Research questions and interview questions relationship

Furthermore, the identified dimensions from the framework of analysis were mapped against the research questions to explore the possibility of refining the research questions. Connection between the research questions and the dimensions can be seen in table 3.4 below.

Research questions	Dimension
Do stakeholders perceive that developing shared understanding problematic (or difficult)? If so, what do they say the barriers are?	Barrier
What do stakeholders do (processes, techniques, tools, representations) in order to develop shared understanding of the problem domain with their client?	Stakeholder, Process, Representation, Tools & Techniques
How does what the stakeholders say they do compare to what is reported in research literature?	All

Table 3.4 – Research questions and analysis framework's dimensions relationship

In parallel with the development of the interview questions, the identification of candidate participants was also conducted. The following section describes the participant selection process.

3.4.4. Participants selection

Potential candidates to invite as research partners were identified from publicly available information, in particular the NZ Business Who's Who database and the MIS 100 magazine. Previous cooperative relationships with practitioners were also leveraged. The criteria used for selecting the candidates were:

- Software development companies, which is the context of the study on the development of shared problem domain understanding.
- Auckland based local organizations, as it would ensure the researcher having face-to-face interviews in order to obtain richer data (Fox, 2006), and avoid the need for travelling.

- Small to medium sized organization, as it reflects majority of the New Zealand market, and may increase the applicability of the findings.

It was felt that the vendor representative to be interviewed should be at a senior level, and ideally be close to the technical developers, because they would be most capable of providing insights and an overview of the process of sharing domain understanding. It was also hoped that the participating organisations would include a range of application domains so that the any relationships between the application domain and shared understanding may be explored. A common distinction made between software development companies is whether they develop largely product based software or (i.e. market-driven) or bespoke software (i.e. software service). It was hoped that a mix of both these types would participate in this study, again in order to explore any relationships with this company type and shared understanding.

A total of 204 candidates were invited to participate in this research. The invitations were mailed along with the form for obtaining participant's consent, as well as participant's information sheets to the IS manager of each organisation. Examples of these documents are available in appendix B, appendix C, and appendix D of this thesis.

From the 204 candidates, 11 organizations accepted the invitation to participate in this research and 10 of these were Auckland based. Representatives from each of the Auckland based organisations were interviewed, with one of the participant representing two different organizations (Organization 7 and 10). All of the responding participants were at a senior level in their respective companies, ranging from senior system analyst to company owner. The range of application domains participating Details of the organizations who participated in the study is presented in table 3.5.

Company	Application Domain	Service type	Interview date	Participant’s Position
1	Museum & collection management	P	November 14, 2007	CEO & System Consultant
2	Industrial & metal machine tool	B	November 8, 2007	Business Owner
3	Document & business process management	B	November 7, 2007	IT Delivery Manager
4	Retail related domain	B	December 3, 2007	Business Owner
5	Printing & graphic industry	P	November 12, 2007	Technical Director
6	Health IS	B & P	November 28, 2007	Project Manager
7	Student management system	B	December 5, 2007	Development Manager & Senior System Analyst
8	Finance & sport club management	B	December 10, 2007; December 13, 2007	Director
9	Varied	B	December 5, 2007	Solution Manager
10	Commercial related domain	B	December 12, 2007	Director

Table 3.5 – Participating organizations (Note: P – Product driven, B – Bespoke)

Each of the positive response from the participant candidates were followed up with a phone call to the corresponding candidates for introduction purpose, as well as making an interview appointment. All of the appointments were made according to the participants’ schedule and preferences to ensure the participants’ comfort. Prior to the interview, the interview questions were sent to the participants, in order to allow the organizations some preparation time and to familiarize themselves with the structure and content of the interviews. Although some of the spontaneity of the answers is lost, there is a likely increase in the depth of the answers if the participants have had a chance to think about the questions.

The selection of multiple companies in this study is intended to identity trends or patterns regarding development of shared understanding activities. In addition, comparing the results for different companies (and their different contexts) may lead to insights for improvements or changes. Furthermore, the selection of multiple companies should increase the possibility of readers finding cases which are relevant to their circumstances, therefore increasing the chance of naturalistic generalization (Stake, 1995, as cited from Tellis, 1997b).

3.5. Conducting the data collection phase

The second stage of conducting a case study according to Yin (2003) is the data collection stage, which in this case is conducting the semi-structured interviews in the field. The actual data collected was in the form of notes and annotations on the question sheet as well as a recording of each of the interviews.

A pilot-case study was conducted with one of the participants prior to starting the data collection stage. This was done because pilots are useful for testing and refining the research instrument, as well as the data gathering protocol (Dubé & Paré, 2003; Fox, 2006; Seidman, 2006; Yin, 2003). In addition, the execution of pilot-case study was also intended to help the researcher to gain experiences and refine the researcher's interviewing skills. During the pilot study, the project supervisor, who also attended the interview session, provided some support by probing some responses that needed clarification and later making some suggestions for improving the researcher's interview technique. The pilot study suggested some minor modifications of the words and sentence structures to improve interviewee understanding. Changes to the protocol were not required.

The same protocol and research instrument were repeated in each interview. According to Tellis, (1997; 1997b) and Yin (2003), this should improve the reliability of the research, as well as maintaining uniformity of the data (Sadraei, et al., 2007), simplifying data analysis. Most of the interviews were conducted at the participants' work places to minimize the interruption of participants' schedule, except for one of the participants who preferred to be interviewed outside his office. The interviews generally took between 1.5 to 2 hours. In some of the interviews, the researcher's project supervisor was present during the interview and occasionally provided additional probing questions.

After confirming the interviewee's consent to record the interview, the structure of the interview and the management of privacy were restated, as suggested by Fox (2006). This was followed by a review of the research background and motivation, as well as a high-level, conceptual description of the phenomenon to be investigated, with an opportunity for the participant to clarify any points. This ensured all participants had a basic understanding of the concepts and terminology used, but minimised influencing their future answers to questions.

Participants were put at ease and a relaxed atmosphere encouraged, which Fox (2006) suggests is important, by having the interviews take place in an environment that the participants are comfortable in, and also not "jumping in" to the interview, but establishing a rapport with the interviewee through relaxed conversation first. The interview questions are structured loosely around the research questions, starting with some demographic questions. Each section started with an open question followed by more directed questions.

As is common in semi-structured interviews (Fox, 2006; Seidman, 2006), the prepared questions acted as a guideline only, and often they were expanded or additional questions asked.

While enquiring, the researcher also made interview notes onto the interview template. Sometime the answer to a question was provided by a participant before that question had been asked (because it is later in the question set on the interview template). This meant the notes were not always taken in sequential order on the interview template. Confirming Seidman's (2006) observations, these non-sequential notes were sometimes useful during the interview because they provided a useful trigger for deeper probing when that topic was revisited following the question sequence on the template.

Fox (2006) warned that the use of an interview template may introduce bias by increasing the chance of interpretation error by the researcher. The researcher may tend to interpret certain answers so that they can be matched to an existing question on the template, even if they might not be a good match. This potential source of bias is mitigated in this study by referring back to the transcripts of the recordings of the interview to check whether correct interpretations were made or not.

Seidman (2006) demonstrates that another important technique in conducting an interview is to ask the participants to reconstruct rather than remember their experience. He argues that asking the participants to remember the experience is more likely to trigger response bias due to their poor memory. Asking them to reconstruct their experience, however, is more likely to elicit what the interviewees thought as important rather than the whole process in details. Following Seidman’s suggestion, interview questions starting with “Do you remember ….” were avoided. In addition, to further avoid the poor memory effect, the questions was designed to systematically target their latest experience first, to help them reconstruct their experience, and followed by a similar question for a more general scope.

A number of experienced researchers (see for example Collis & Hussey, 2003; Fox, 2006; Seidman, 2006), as Collis & Hussey (2003)) point out that the interviewee should be aware of non verbal cues which can enhance the interpretation and confidence of the data gathered. The researcher thus tried to be aware of the occurrence of non-verbal cues during the interviews, such as body language and voice intonation, and use them as useful triggers for asking deeper information or noting emphasis and mood.

3.6. Data analysis

Data analysis follows the data collection stage and in this section, the unit of analysis, the method of analysis and generalization of the results are all discussed.

3.6.1. Unit of analysis

One of the fundamental elements of case study research is the unit of analysis, which defines major entity being investigated in the case study (Yin, 2003). Moreover, as pointed out by (Markus, 1989 as cited from Dubé & Paré, 2003), the unit of analysis provides information about the boundaries and scope of the case study and is therefore useful in distinguishing which situations the results may apply to.

In this research, the process of developing a shared understanding is selected to be the unit of analysis. To be more precise, the process is limited to the context of vendor and client stakeholder groups sharing understanding of the problem domain in software development. As Tellis (1997b) argues, analysing a process may provide a holistic view of a system of action. In the case of software development Sadraei et al. (2007) list the system of action as including activities, artefacts, roles, tools, and techniques. All of these elements match the research questions which suggest that this unit of analysis is well justified. The next section describes the method of analysis used to analyse the interview data.

3.6.2. Method of analysis

The data collected from each organisation are analysed mainly at the organisation level and qualitative results are emphasised because of the small sample size.

The method of analysis employed in this research is a thematic analysis of the interview data, which includes the notes taken during the interview together with a full transcript of the interview. This is one of the common methods for analyzing interview-based research in order to identify concepts or themes related to a phenomenon (Fox, 2006). Similar approaches can be seen in studies, such as Ojasalo et al. (2007), Robertson (2007), and Tsui & Gururajan (2007). Moreover, in its implementation, thematic analysis does not need any fixed or pre-defined codes/categories for conducting the coding process. This feature is considered appropriate in this research, since it provides more opportunity to identify new insights, as well not limiting the data analysis to using any pre-defined codes which may not be a good fit for the data (Seidman, 2006), thus reducing potential bias.

Thematic analysis is similar to content analysis in many ways, with one major exception, content analysis involves quantification of the data, whereas thematic analysis doesn’t (Marton, 1986). Thematic analysis also has some commonality with the data analysis in the Grounded Theory method. The main difference between thematic analysis and Grounded Theory is that the latter puts emphasis on incorporating existing theories into the analysis process (Braun & Clarke, 2006), which is not in the case for thematic analysis.

The analysis phase started with transcribing the recording of the interviews, followed by review of these transcripts to gain strong familiarity with the data. The transcript content was firstly coded by identifying passages that related to specific points (at a higher level of abstraction than the transcript content) and these became themes as patterns emerged. For example, the point of “an uncooperative client representative as a barrier to shared understanding” emerged as a theme coded as “uncooperative client” within the case organisations, since narratives related to this repeatedly appeared in the interviewees’ transcripts. This coding process was supported by the use of Nvivo 7 software, which helped with code management, retrieval of coded passages, and the classification process.

After completing this initial thematic coding process these themes were then categorised into the factors from the framework of analysis depicted in figure 2.5 if there was a “natural” fit. This allowed the usefulness of the framework of analysis to be evaluated by checking that all the identified themes could be categorised plausibly within the dimensions of the framework of analysis. Otherwise, the model represented by the framework of analysis would need to be extended or modified. Also this framework is the basis for comparison of the interview data with literature. For instance: a passage from the transcript which was coded as “uncooperative client” could reasonably be placed in the “barrier” dimension of the analysis framework. During the coding process the link between a coded excerpt of the transcript and its location in an organisation’s original transcript was handled by utilizing that feature in Nvivo 7.

The process of categorizing the coded themes was done in iteratively until all of the codes were categorized. The iterative process also included:

- Ensuring that the coded passages were categorized based on their real context, in order to avoid alteration of the real meaning intended by the interviewees, as suggested by Bowden (2000). As de-contextualizing the data may potentially decrease validity of the result, it was important to refer back to the transcript to check this whether the passages coded according to their context or not.
- Ensuring that the labels given to the passages were unique, consistent and matched the themes of the excerpts. This process sometimes resulted in new labels or changing an excerpt’s labels or even breaking the labelled passage into several new passages and labelling them

In order to ensure that the transcripts were properly coded and categorized, researchers (e.g. Bowden, 2000; Boyatzis, 1998) argue that more than one person should be in volved in the coding and categorizing of the themes. This was achieved to some extent in this study by having discussions regarding the coding with the project supervisor and other colleagues.

Following the completion of the categorization process, the researcher further anlaysed the categorised data through the following activities:

- Identify similarities and differences between the case organisations and look for possible trends or explanations of differences, as suggested by Fox, (2006).
- Identify passages that were emphasised and were of obvious special importance to the interviewees. This might be because they are told in a striking manner (e.g. particular passion, excitement, enthusiasm, animation, or even indignation) or emphatic language. Such passages were noted during the interview as part of a response. However, researcher need to be aware of participants bias in handling such passages (Seidman, 2006)
- Identify similarities and difference between the findings of this study and what isreported in literature (Seidman, 2006).

The results of the data analysis are presented in Chapter 4.

One of the common issues discussed in literature is the generalizability of the results from case studies. This has important implications for this study and so several points of view are presented and discussed in the next section, and related to the research in this thesis.

3.6.3. Generalizability of the result

Yin (2003) in his definitive book on case study research discusses the traditional prejudice that views a single case study (or a few) as not generalizable to the wider universe. In response of this, several researchers have developed arguments that take a contrary view and broaden the notion of generalizability.

In his argument regarding this matter, Yin (2003) stresses the difference between 'statistical generalization', which occurs from experimentally-based research and 'analytical generalization'. The latter focuses more on comparing a previously developed theory to a research result, in order to enhance the generalizability of that theory. One of the aims of the research in this thesis is to compare and contrast literature, represented by the framework of analysis and the current understanding of its dimensions, with practice and experience uncovered from the multiple case studies. This presents an opportunity to support and/or extend and/or or challenge current literature related to shared domain understanding from the case studies, in line with Yin's concept of analytical generalization.

From interpretive case studies point of view, Walsham (1995b, as cited in Tellis, 1997b) argues for four types of generalization that can be relevant to case study research, namely: the development of concepts, the generation of theory, the drawing of specific implications, and the contribution of rich insights. It is the aim of this research to gain some richer understanding of the development of shared understanding and any barriers. This should have specific implications for practitioners in terms of process refinement and support.

Another point of view is presented by Stake (1995, as cited in Tellis, 1997b) and Seidman (2006) who maintain that the generalization of the results of a case study depends largely on a reader's interpretation from their experience, in the sense that something about the results of a case study may "resonate" with the reader's experience. They claim that this will often result in a better understanding of the phenomenon which is the topic of the case study, and creates what Stake calls "naturalistic generalization. This research provides an opportunity for naturalistic generalization through the practitioners and researchers who read it.

3.7. Conclusion

This chapter positions the research in this thesis in the spectrum of approaches described in literature and justifies selecting a specific approach and methodology in terms of demonstrating that the selection is well suited to the nature of the research questions and aims. Within this wider research methodology, the use of a multiple case study approach is justified and the semi-structured interview as a data collection method is also rationalised. The data analysis methods to be used are described and it is shown that these will enable the research aims to be realised. This chapter also includes details on the implementation of the selected research methods and techniques, showing how the protocols used contribute to data reliability, integrity and generalizability..

In the next chapter, the result and findings of this multi-case study research are presented, along with some relevant discussions and linking with the research questions.

Chapter 4

Research Findings and Discussions

4.1. Introduction

In order to obtain depth in understanding of current practices and issues in developing shared understanding, as well as their correspondence to current literature, the data obtained from the interviews were coded, categorized, and analysed. The framework of analysis developed from the literature review, depicted in figure 2.5, is used to provide a focus for identifying patterns of practice in sharing understanding. The analysis of the data includes an inter-organisational comparison of practice, a demographic analysis, as well as comparison with existing literature.

Result of this analysis process is presented in this chapter in the form of research findings and discussion of relevant interesting aspects of these findings. The findings, analysis and discussions presented in this chapter are the basis for developing the conclusions of this research, answering the research questions, as well as revealing possible future research directions.

In the next section the demographic data of the participating organisations are presented and discussed. This is followed by the research findings, analysis and discussion section (section 4.3). This comprises several sub-sections structured around the dimensions of the framework of analysis presented in the end of chapter 2 (figure 2.5). Continuing this section is an overall discussion section which presents some interesting themes that emerge during the analysis of the result (section 4.4). Finally, the last section (section 4.5) summarises the main conclusions and points from this chapter.

4.2. Demographic

In a multi-case study approach there should be some similar characteristics in the selected cases in order to ensure that the comparison of the cases along these similar dimensions is valid. For example, the aim was to select companies whose (a) main business is software development (and thus share understanding for requirements engineering), (b) would be classified as Small to Medium Enterprises (SMEs) in general (<100 full time equivalent staff is most relevant to the NZ context), and (c) are well established in the New Zealand market (established over 5 years ago and so have informed processes). Thus organisations sizes (staff number), the longevity of the organisations and the project activity levels were included in the demographic data gathered.

Another aim was to see if there were variations in practices and issues among different business domains and so business domain was selected as a demographic to collect. From literature it is not clear if shared understanding practice is different for bespoke (software service) versus product-driven software development and so this was also collected as a demographic. The experience and role of the interviewees may also affect the data gathered about shared understanding, so this was included in the collected demographic data.

Section A.3 of the interview questions has the specific demographic questions asked. In addition there are also role-related questions which are intended to elicit the interviewees' roles in the context of vendor and client relationships and also the interviewees' roles in supporting the development of shared understanding between the vendor and the client groups.

The collected demographic data are summarised in Table 4.1. It shows that there is considerable diversity in the business domains represented by the participating organisations as desired.

Org	Bus. Domain	FTE	Longevity	Project	Role and Expertise	Experience	Business
1	Museum & collection management	10-20	22	5	CEO &System consultant (2) (Technical & Domain)	> 10 years & 9 years	Product driven
2	Industrial metal machine & tools	<10	17	4	Owner (Domain)	25 years in process controlling	Bespoke
3	Document & business process management	21-49	11	4-15	IT delivery manager (Technical)	4 years	Bespoke
4	Retail related domain	<10	21	~ 6	Owner (Domain and technical)	> 10 years	Bespoke
5	Printing & Graphic Industry	10-20	18	1 (major release)	Technical Director (Technical)	> 10 years	Product driven
6	Health Information System	50-99	21	120-150(mostly small)	Project Manager (Technical)	> 10 years	Product driven and Bespoke
7	Student management system	10-20	10	3 (major) +20-30 (minor)	Development Manager & Senior Sys. Analyst (2) (Domain and technical)	> 10 years	Bespoke
8	Finance & Sport club management	<10	11	3	Director (Domain)	None	Bespoke
9	Varied	21-49	18	~ 20	Solution Manager (Technical)	> 10 years	Bespoke
10	Commerce related domain	20	8	~40	Director (Domain and technical)	> 10 years	Bespoke

Table 4.1 – Demographic data for the case organisations

The sizes (FTEs) and ages (longevity) of the organisations all conform to the objective of investigating established SMEs. The interviewees' roles are all at a senior level although there is a mix of domain and technical expertise. All but two of the interviewees have at least 10 years of experience in the software development industry, with the other two having considerable domain and business experience.

Additionally, from the interviewees' transcripts, information about the company business type is revealed. Seven out of ten organisations can be categorized as bespoke type software development companies since they deal with custom-made products designed exclusively for clients who requested them. Two from the remaining three organisations are product driven companies which produce mass market products, while the last organisation covers both type of business. The differences in requirements elicitation and verification processes, and related shared understanding, are apparent between the two types of business models, as discussed in the next section.

In the next section, the research findings, analyses and discussions are presented.

4.3. Research findings, analysis and discussions

The research findings of this study are based on interviews at 10 software development organisations in New Zealand. These were the 10 organisations (out of the invited 205 organisations) who agreed to partner in this research investigation. Although the invitations were addressed to the IT Manager of the organisation, the interviews were conducted with self-selected senior representatives from each of the organisations and ranged from senior system analyst to company owner.

The following sub-sections are based on the dimensions of the framework of analysis developed in chapter 3 (figure 3.1). In each framework dimension, the interview transcripts are analysed and coded to identify themes and patterns. Their implications for practice and current research are discussed. In section 4.3.1 the barriers to developing shared understanding are analysed and 11 themes identified, which are further grouped into barriers related to stakeholders, communication and representation. The stakeholder involvement dimension is analysed in section 4.3.2. , Process of developing shared understanding (section 4.3.3), Tools, techniques and their efficacy (section 4.3.4), Representation of understanding (section 4.3.5), and Perception of importance in developing shared understanding (section 4.3.6).

4.3.1. Barriers to developing shared understanding

The first research question investigates practitioners' perceptions of whether developing shared understanding is problematic or not, and if so, what they think the significant barriers are.

A problem refers to something unwelcome which needs to be overcome. In the context of developing shared understanding, the presence of various barriers can hamper the participants from achieving a desired level of shared understanding. Therefore, the existence of barriers can be considered as problematic and a better understanding of these barriers should result in better designs of processes and techniques for overcoming them, and ultimately better quality software.

Findings on the barriers are based mainly on transcripts of interview questions 19 to 28 (see Appendix A). The participants were initially asked an open question about barriers in sharing the business knowledge with their clients for their most recent software development project. They were asked to identify what barriers they considered significant, how they addressed those barriers, and how effective they were at overcoming the barriers. Three specific barriers identified from literature as being significant were then investigated explicitly through questions 23 to 28. These are (1) the use of specialised terminology or jargon, (2) managing the evolutionary nature of shared understanding, and (3) managing multiple (possibly opposing) points of view. Table 4.2 presents a summary of the 11 main themes of barriers that participants' identified, as well as their frequency among the participating organisations.

	Barrier	**Frequency**
Stakeholder related barriers	*Inadequate client representatives*	**10**
	Diversity between stakeholders groups	**8**
	Change in problem understanding	**7**
	Client uncertainty	**6**
	Client's internal conflict	**4**
Communication related barriers	*Lack of common language/terminology*	**8**
	Difficult access to key stakeholders	**8**
	Communication timing and frequency	**3**
	Lack of enough "rich" communication	**2**
Representation related barriers	*Unfamiliar or complex representations*	**5**
	Ambiguity of natural language	**4**
	Non-engaging representations	**3**

Table 4.2 - Result on barriers in development of shared understanding

Interestingly there is unanimous agreement among the participants for only one barrier: the quality of the client representative(s). This was consistently identified in the open interview question about barriers and the prevalent view expressed was that this is a key interface to the client organisation and if this relationship and interaction is poor, or if the domain knowledge of this representative is poor, this can be a significant barrier to evolving a shared understanding between the two groups.

Unsurprisingly the three barriers explicitly referred to in the interview (change in problem understanding, lack of common language, and disagreement in understanding) have high frequencies (although not unanimous agreement that they are barriers). More surprising was the observation that these barriers were generally not identified in the open question phase and seemed to have less “front-of-mind” significance than the quality and accessibility of key stakeholders.

The significance of this and possible implications for practice, as well as the meanings and significance of the other barriers are discussed in more detail the following sub-sections.

In table 4.2 the 11 barriers identified are further categorised into three high level barrier types that align with some of the other dimensions of the framework of analysis identified from literature. These are, in descending frequency order: (1) stakeholder-related barriers, associated with the level of positive engagement and useful domain knowledge of the stakeholders; (2) communication-related barriers, associated with issues around the quality and timing of communications and related processes; and (3) representation-related barriers, associated with issues to do with a documented representation of the domain understanding being perceived as a key mechanism for sharing and verifying this understanding.

The high level classification of ‘*stakeholder related barriers’* was the most frequently discussed area among the participating organisations. Its dominance is shown by the variety of barriers under this category, as well as their frequencies across the participating organisations.

In terms of communication, significant emphasis is given by participants to the barriers caused by the use of jargon in communication and also the difficulty in accessing the key stakeholders, who tend to have the desired knowledge, skills, or capabilities to be shared.

Another barrier categorised under “communication related barrier’ is inadequate timeliness or frequency of communication activities between the vendor and client groups.

Under ‘representation related barrier’ category, the barriers are mostly due to the employment of advanced representations, typically graphical (e.g. UML). The views expressed generally related to the barriers which involved use of representations which were unfamiliar to the clients. Typically this barrier was related to lack of clients’ perceived value of using a particular representation in terms of the improvement in shared understanding versus the high (time) overhead to understand the representation (e.g. representation with high degree of formality). Another point raised as a barrier to shared understanding concerns the ambiguity of natural language and the consequent likelihood of unrecognised misunderstandings. A few of the participants also indicated that they perceived certain representation as being “dry” and not engaging (e.g. “thick and dry documentation”), which they claimed could also hamper the development of shared understanding.

In the following sub-sections, each of the categories presented in table 4.2 are discussed in more detail. This includes a discussion of the dimensions of the barriers, comparison to existing literature, as well as suggestions for handling or overcoming the barriers.

4.3.1.1. Stakeholder related barriers

The barriers which are categorized into ‘stakeholder related barrier’ category concern stakeholders’ level of engagement (e.g. uncooperative, passive, unforthcoming, etc.), quality (e.g. level of knowledge) and other particular stakeholders’ circumstances that can hamper the development of shared understanding between the vendor and client groups (e.g. multiple point of views among the client stakeholders).

The barriers under this category are: inadequate client representatives, inter-stakeholders group diversity, change in problem understanding, client uncertainty, and client group internal conflict (see table 4.2). From all of these barriers, ‘inadequate client representatives’, ‘inter-stakeholders group diversity’ and ‘change in problem understanding' seem to be more significant compared to the others based on the number of organisations which perceived them as barriers in developing shared understanding.

A. Inadequate client representative

The barrier 'Inadequate client representative' includes poor client attitudes to sharing understanding, as well as their level of domain knowledge and involvement in knowledge sharing activities. Table 4.3 details the dimensions coded from the transcript analysis that are included in this barrier category.

Barrier	Dimensions from Transcript	Number of organisation
Inadequate client representatives	*Client's lack of domain knowledge*	5
	Uncooperative client	5
	Passive client	2
	Client with hidden agenda	2
	Unforthcoming stakeholders	2
	Overly demanding client	2

Table 4.3 - Dimensions of inadequate client representatives

The inadequacy of the client representative is basically related to either an inadequacy in ability to satisfy the knowledge needs of the vendor, or as an attitudinal or behavioural problem that made knowledge sharing interactions "hard work" or "untrustworthy".

A.1 Lack of domain knowledge

Having client representatives which did not have sufficient domain knowledge to address the vendor's knowledge needs or did not have a strong capability to articulate or analyse their knowledge, was identified by half of the case organisations as a significant barrier to shared understanding. The interviewee comments in this area are typified by organisation 6's description:

> *Sometimes the person [client representative] doesn't have the knowledge they need to have. All they might have is strong ideas…but haven't canvassed that idea with other users*

The consequences of this barrier were generally described as unnecessary uncertainty or volatility in requirements, where the client representative “changes their mind every five minutes” (organisation 5).

The lack of domain knowledge of the client representative(s) is identified in literature as a possible issue in requirements engineering (Cao & Ramesh, 2008; Emam & Madhavji, 1995; Lu et al., 2007; Shin & Sutcliffe, 2005) but isn’t given the degree of emphasis that was apparent from the five practitioners in this study who discussed it. This suggest that development of techniques that increase the likelihood of negotiating client representatives with appropriate domain knowledge would be of significant interest to practitioners and should improve the development of shared understanding.

Even though lack of domain knowledge appears to be one of the main areas of client’s inadequacy, no clear explanation regarding why this situation occurs was presented by the interviewees. Why should such individuals with insufficient domain knowledge be assigned to represent the client group? Coughlin et al. (2003) suggest that this is a consequence of “stakeholders [being] chosen on the basis of their position and status rather than their knowledge per se”. Al-Rawas and Easterbrook (1996) in their empirical study conclude that one underlying causes of this situation may be the vendors’ lack of involvement or lack of authority in selecting the client representative, typically selected solely by the client group. Furthermore, according to Emam and Madhavji’s empirical study (1995), the risks of inadequate client representation are higher in “small and rapidly growing organizations where the most capable users are not available”. None of these reasons, however, can be confirmed from the interview results of this study.

In term of handling this kind of barrier, a variety of approaches are identified in the participants’ responses. One of the participants (organisation 5) emphasized the importance of stakeholder selection process in order to prevent the barrier of client with insufficiency domain knowledge by suggesting the selection of stakeholders who are ‘forward looking’ and have a clear vision of the organisation’s desired future goals. Another organisation had the view that it is unlikely they will have the opportunity to select the desired client representative and they just had to make the best of this situation:

> *…you generally can't go behind that person's back and seek refinement. You have to work through that person the best way you can. –* ***Organisation 6's Project Manager***

Another approach to overcoming a client representative's inadequate domain knowledge is for the vendor to provide domain knowledge from their previous experiences and to guide and inform the client on the business domain. This was suggested by organisation 3's IT delivery manager as well as being discussed in the literature (e.g. Emam & Madhavji, 1995). However, this approach is obviously limited to vendors who have domain expertise available in their group. Moreover, the effectiveness of this particular approach (providing consultancy or suggestions) also depends heavily on the client's willingness to accept the vendor's advice.

A.2. Attitudinal or behavioural problems

In addition to a client representative's lack of domain knowledge, attitudinal or behavioural problems were also identified by the participants as contributors of a client representative's inadequacy. These attitudinal or behavioural problems are:

- **Uncooperative client**, which refers to a client representative's unwillingness to do something asked by the vendor group.
- **Unforthcoming client**, which refers to a client representative's unwillingness to provide the knowledge essential to the project.
- **Passive client**, which refers to a situation where a client representative gave little or no contributions to developing shared understanding and often just agrees with everything.
- **Client with hidden agenda**, which refers to a situation where a client representative brought up his or her own idea or agenda, without socializing it to the other client group members at the first place.

- **Overly demanding client**, which refers to a situation where the client representative was being unreasonable or tended to force unrealistic expectations to the vendor.

Uncooperative client behaviour is identified by half of the case organisations. The participants described this lack of cooperation as generally showed up during the process of verification and validation of understanding between the vendor and client. The interviewees indicated that this hindered the vendor from verifying with the client that they have a common understanding and could result in lack of “buy-in” from the client representative. A typical situation regarding this particular problem is clearly illustrated by organisation 10 where a client representative selected to be involved in user acceptance testing was unwilling to provide any feedback to the vendor’s BA because she hadn’t been involved in specifying her requirements earlier in the project (at the discretion of the client organisation). This lack of cooperation held up the user acceptance testing and was a barrier to sharing understanding.

Interviewees described the consequence of these poor clients’ attitudes or behaviours as resulting in extra effort or heightened project risk. This included extended effort to probe the client representative for clearer understanding, to develop trust and a more collaborative relationship, or expanded negotiation. In some cases, despite these efforts to get agreement about understanding with the client representative, this agreement or “buy-in” was not always achieved. This is exemplified by organisation 7’s experience where they describe that even after considerable negotiation and consultation with the client, resulting in what the vendor thinks is an accurate representation of their shared understanding, “9 times out of 10” they don’t get requirements signoff from one or more relevant client stakeholder. The interviewee describes their usual reaction to this is “to go forward with the project without full signoff”, accepting the increased risk.

Communication barriers related to attitudinal or behavioural problems have long been identified in past literature and are still identified as problematic to this day. For example, Emam and Madhavji (1995), in their empirical study on RE practice, identify a lack of cooperation between IS groups and business groups as a result of “personality clashes” as a barrier to communication and shared understanding. Also Saiedian and Dale (2000), in a review of requirements elicitation techniques and issues, identify several similar attitudinal problems as barriers to shared understanding between customers and developers. They describe passive and compliant client behaviour as forms of client resistance (“silence resistance” and “compliance resistance” respectively). More recently, Cao and Ramesh (2008), in their empirical study on agile requirements engineering practice, identify clients with unrealistic expectations as a barrier to shared understanding.

It has been suggested (e.g. Emam & Madhavji, 1995) that stakeholders’ personalities may be at the root of some of these attitudinal or behavioural problem. Although personality was not expressed explicitly by interviewees in my study, they did identify a number of client personality traits that disrupted communication and understanding. This included client representatives who are selfish, thinking only of their own gain (organisation 3), and clients overly secretive, who are unwilling to release information they inappropriately viewed as commercially sensitive (organisation 2).

Another factor identified in literature that may trigger attitudinal or behavioural problems is a client stakeholder’s resistance to the impact or changes that might be brought by the introduction of a new system (Saiedian & Dale, 2000). In my study this was illustrated well by organisation 3’s participant. He describes the negative (“terrible”) feedback on a newly implemented system and the subsequent uncooperative behaviour of a client. It turns out that the new system made it more difficult for the client (a doctor) to have eye contact with his patient and this was unacceptable to the client.

The participant from organisation 10 describes an interesting scenario that resulted in overly demanding client behaviour and a barrier to sharing understanding because the scope of the understanding kept changing. In this scenario the vendor is contracted to build a web site for a client and showed the client some existing websites with similar functionality. The client continued this approach of scanning websites for ideas even after development had started. She kept coming up with new requirements based on these other web sites, expecting that it would be easy to incorporate these ideas into the site being developed. Her unrealistic (changing) demands and expectations caused frustration with the vendors and became a significant barrier to communication. The interviewee went as far as arguing that “it’s dangerous in certain circumstances to use a prototyping approach. You have to pick the right client”. A similar view is expressed in some literature also (for instance: Cao & Ramesh, 2008; Emam & Madhavji, 1995).

In terms of addressing these attitudinal or behavioural barriers, the interviewees had few suggestions. One of the suggestions identified by organisation 3 concerns the need for the vendor representative to be “reasonable and open-minded” when “taking feedback” from the client representative. This emphasises the importance of for selecting vendor representatives with good communication skills. This is also pointed out in research literature (e.g. Curtis et al., 1988; Emam & Madhavji, 1995). In the worst case, one of the case organisations (organisation 10) suggests to seek support from a higher authority in the client organisation to deal with their problematic representative. This particular approach suggests that there is an advantage for developing good relationships with top management on the client side.

B. Inter-stakeholder group diversity

One of the major barriers to developing shared understanding that was identified by almost all the participants is linked to the differences in backgrounds between people in the vendor and client groups. These differences in backgrounds between the groups result in differences in individuals’ characteristics such as experience, knowledge and level of knowledge. These in turn result in differences in group members’ values, world-views, trust-building, risk tolerance, priorities, ability to conceptualise and so on. The interviewees tended to attribute noticeable or significant difference “trends”, built up over a series of inter-group interactions, to the groups themselves. They described these differences as potentially disrupting communications (and shared understanding) between the groups. They noted that this could lead to misunderstandings, misinterpretations, or miscommunications between the groups. This, in turn, could lead to unexpected actions, behaviours, or decisions, and will possibly result in increased levels of frustration or conflict, inefficient processes, the need for re-work, poor quality systems, or even project failure.

The differences in domain knowledge, typically more technical in the vendor group and more business oriented in the client group, are highlighted in the following quote from the participant from Organisation 10.

> *'… because with the developer and the client it is sometimes very difficult to get them to speak the same language … some projects failed badly, because the developer couldn't comprehend the business process at all …*

This participant went on to describe the problem of developers "jumping into the coding process" before they understand the business processes and goals sufficiently and described the result as:

> *…a big cycle of change, change, change, change, and change, you never finished and you delivered late, you delivered poor quality …*

In order to alleviate the problems caused by differences in experience and knowledge, two organisations employed domain experts, both previously in a client role, with experience in information systems and relevant business domains. Lubars et al. (1993) notes this approach in his study also. The domain experts act as a liaison or "bridge" between the vendor and client groups. They have an understanding of the domain knowledge shared by the clients as well as an understanding of the needs of the vendor's development group and transfer the knowledge in a more "palatable" fashion to each group.

Another identified difference that can hinder the development of shared understanding is the difference in knowledge level possessed by the two communicating people. This particular situation is clearly depicted in the following quote:

> *'…other barriers, things like understanding where the business is trying to go and that usually happen with lower level staff … you know if they don't understand what the business are trying to do, they can get lost, you know they hear all this people talking about things that they don't understand and they don't see the point of it.'*

The above comment indicated that different level of knowledge on the business aim could result in communication gap between the two groups and certainly it could affect the remaining knowledge sharing process. In other cases, interviewees expressed their struggle with project members' turnover in the middle of a project, which resulted in similar situation when there are new members joined. In this situation, the interviewees who expressed the problem agree about the necessity to refer to documentation at this situation; however one of them argued that relying on documentation only is not sufficient for getting a clear understanding.

This issue implies that there is a need for a well-established knowledge transfer scheme for and between the two groups. Some of interviewees also spoke of their efforts in handling this barrier by facilitating a gathering for involving stakeholders, such as a project briefing meeting, focus group, or other similar activities to ensure the knowledge sharing process happened among involving stakeholders. However, it seems that this approach is more suited for preventing such a barrier. In case that it happens in the middle of the project, it seems there is no better way so far than ad-hoc knowledge transfer to fill the knowledge gap, for example by contacting key person (e.g. ex-project members) as one of the organisation usually do.

Literature seems has not mention this barrier yet or maybe not explicitly, as this probably because of this barrier is closely connected to poor communication or regarded as the result of poor communication.

One interviewee found that having either a client representative or vendor representative who did not have English as their first language was a significant barrier to communication and developing shared understanding. This participant went on to note that they lessened the problem by presenting the knowledge in a more visual way (e.g. using pictures, prototype, etc.) where possible.

It is interesting to notice that almost no organisation mentioned differences related to culture as a barrier, despite it being commonly identified in literature as a significant barrier (e.g. Damian, 2007; Hsieh, 2006). One possible explanation for this could be that the case organisations may only do business in their local market, with little cultural diversity.

C. Changing problem understanding

Changing or evolving problem domain understanding during software development is well documented in literature (e.g. see Curtis et al., 1988; Hilton, 2003; Lawrence, 1998; Saeidian & Dale, 2000). The literature suggests that such changes need to be communicated and a shared understanding of the change, as well as its impact, need to be developed. Typically new understanding results in changes to requirements and needs managing to integrate into the project positively. Literature reports that this can be problematic.

In this study no case organisation identified changing understanding as a barrier to developing shared understanding until prompted by the interviewer. However, when prompted, most participants agreed that evolving understanding of the problem domain is normal and that if not well managed can be problematic.

Participants viewed changing understanding as generally positive, ultimately resulting in a better quality system. They described the need to refine understanding as often being triggered by the presentation of new ideas or questions that challenge the client group's current understanding during communication between the vendor and the client groups.

They described their experiences where the changes in understanding were problematic when:

- Changes to understanding were overly frequent, resulting in uncertainty and requirements instability;
- New understandings developed by the client group are not communicated to the vendor or developer group;
- The client is not clear on the wider implications or impact of new understanding.

This barrier has been classified as a stakeholder related barrier because it could be argued that the problems identified by the interviewees can be attributed to the client characteristics (respectively):

- Client uncertainty or inexperience in the problem domain
- Unforthcoming clients
- Clients have insufficient (understanding of) domain knowledge

This suggests that addressing the stakeholder related problem would potentially lessen the barrier related to changing problem understanding. This strengthens the notion that the process of stakeholder selection is important to the success of developing shared understanding between the vendor group and client group.

D. Client uncertainty

More than half of the participating organisations indicated that client uncertainty as one of the barriers in developing shared understanding. Client uncertainty refers to the situation where the client is uncertain about the problem domain or the envisioned system goals, hampering the development of clear shared understanding. According to the interviewees, this could be due to:

- The solution approaches are still at the concept stage and are not well defined.
- Clients' lack of understanding of their own problem domain (e.g. objectives, business process, etc.).

In handling such a barrier, literature suggests the use of a prototype can help the clients to clarify the problem domain and their needs (Berzins et al., 2000; Emam & Madhavji, 1995; Nuseibeh & Easterbrook, 2000). This approach has been applied by most of the participating organisations and interviewees generally reported the use of prototypes as assisting with refining and sharing understanding of the application domain, as well as understanding possible solutions.

Two participating organisations noted that they had been in situations where they had gone as far as ceasing work on a project and offering consultancy services to clients when they realised that the clients lacked sufficient understanding of their problem domain to even discuss possible solution approaches.

E. Internal Client conflict

Several of the participating organisations reported that internal client conflict can interfere with sharing understanding. They described internal client conflict as the situation where there are multiple conflicting or inconsistent points-of-view within the client group and low “overlap” in aspects of their understanding. This lack of shared understanding of the application domain within the client group may not even be recognised by the client stakeholders, who are sometimes unaware of each others’ differing viewpoints and understanding. The interviewees describe two main scenarios for resolving this. The first is where the negotiating activities to achieve compromise and agree on problem domain understanding within the client group may take place solely within the client group in their own time frame and be communicated via the client representative to the vendor. The feeling was that this can be quite time consuming and may be overly influenced by the client representative’s agenda or view point. Organisation 7’s representative describes how a user group acted as a steering committee and “filtered” requirements. This forced them to achieve a level of shared domain understanding so they could agree on user requirements which were to be passed on to the vendor.

> *….the user group decides on whether or not it [the original request from the users] should proceed even, so it may not even proceed based on what they decide internally.*

The other scenario described is where the vendor may be involved in facilitating a group session(s) with the clients to help resolve inconsistencies in client understanding. This was viewed as being quicker to share understanding but did require good facilitation skills.

Most interviewees agreed that it is preferable if the clients agree on the problem or goal (“achieve buy-in”) before they share the particular problem to the vendor. However, findings indicate that in some of the organisations, such an ideal situation is quite a challenge.

The representative from one case pointed out that sometimes the clients went too far in his opinion, and specified aspects of the solutions that were not necessary well founded on a clear understanding of the problem domain. He describes the situation where the clients even restrict the vendor's access to the requirements detailed by the client, and inhibit the sharing of understanding, because the problem domain is not even up for discussion:

> *So they* [client stakeholders] *might gather and write a requirements specification and the IT people will go through that. So I can't have that* [the requirements specification]*, I can't even ask for that. It is usually happens like this: This is the technology that you must use no matter whether it is the right technology or not. So, it seems to be increasingly that we got shut from the client side.*

If the client conflict is not resolved, it can affect the quality of shared understanding, as the vendor can get confused by the conflicting or inconsistent understanding or shared knowledge that does not reflect the need of part of the client group

In literature such a situation is well acknowledged by researchers (e.g. Alcázar & Monzón, 2000; Saiedian & Dale, 2000) and also often identified in empirical studies (e.g. Cao & Ramesh, 2008; Lubars et al., 1993). However it seems there are not many indications of effective way for dealing with this barrier. A recent empirical study (Cao & Ramesh, 2008) suggests that vendors should be proactive by facilitating a "buy-in" process (e.g. forcing a physical meeting for conflicting clients), however, perhaps not all vendors have sufficient authority over the client group to insist on this.

4.3.1.2. Communication related barriers

Another significant theme identified from analysing the interviewee transcripts relates to barriers to high quality communication between the vendor and client groups, needed for sharing understanding. As summarised in Table 4.2, this communications barrier theme relates to four underlying analysis category codes: (1) the use of specialised language whose meaning isn't shared by both groups (jargon), (2) low accessibility of key communication partners in the client group, resulting in missed or less rich communication, (3) the time dimension of communication activities, for example were not timely, were too infrequent or were too brief, and (4) the modes of communication did not allow sufficiently rich communications (e.g. not enough face-to-face communications as often as desired by the vendors).

The second communications barrier, low accessibility, in some respects can be at the root of barriers (3) and (4) but has been kept separate because it was quite strongly identified by the interviewees.

The next subsections discuss more fully the meanings, causes and possible techniques for addressing these barriers, as well a comparison with current literature in this area.

A. Use of jargon

During the open questioning phase of the interview, only two case organisations identified the use of domain-specific jargon or lack of a common language as a possible significant barrier to developing shared understanding. When prompted, another six agreed that this could be a barrier to communications and generally described a technique they used for managing this issue (e.g. maintaining a glossary).

Based on the results of the interviews, three possible situations that this barrier can be situated in were identified by the interviewees: (1) from client to vendor, (2) from vendor to client, and (3) within the vendor group, although for this study we have probed more into the client-vendor interactions.

The typical situation described by the participating organisations is where the terminology is used by the client in sharing knowledge with the vendor but the vendor is unfamiliar with this "jargon". Some of the organisations simply handle this situation by doing iterative clarification with the client side in order to obtain the correct interpretation of the jargon, while some of the organisations handle this situation by employing domain experts, who are themselves familiar with the terminology, to deal with the client. Furthermore, one organisation relied on personal research (e.g. through internet) to anticipate possible jargons used within the business domain.

In term of sharing understanding of jargon, interestingly, only few of the participants stated that they employed a glossary, despite of it being recommended in the literature (e.g. Alcázar & Monzón, 2000; Offen, 2002). Even those that used a glossary didn't find it all that effective because it may be too difficult to create a comprehensive one.

> *'Yes, we have got glossaries, however it doesn't have everything in it. So how do we avoid it [jargon as a communications barrier]…the glossary and I guess education and training as well….'* – ***Organisation 7's Development Manager***

Some of the organisations faced the vendor to client situation, where the vendors have to use jargon that related to the service provided, for example: technical jargon on product, method. For this situation, the most common way based on the interviewee's experience is by giving more effort in explaining the jargon, either by using simpler language or even forcing the adoption of particular terminology.

Finally, the third situation is related to the communication within the vendor group. This problem usually occurs if there are new members joining the organisation. Interestingly, in this study, this type of situation is identified only by product-driven organisations and all of the relevant organisations agreed this can be addressed by internal training or a knowledge transfer program.

The two organisations that did not identify the use of jargon as a barrier had very strong domain expertise within the organisation, and this may be a factor in their response. For example organisation 8's participant answered the question about jargon as a communication barrier with:

> *'Not for me, because I used to work in financial services, I think for some, it's probably yes, because as I say I tend to specialize in the area that I've already worked, …*

B. Difficult access to key stakeholders

In this study, the majority of the participants mentioned that they had difficulties in accessing key stakeholders from the client group and that this resulted lower quality communication. The participants identified a variety of reasons for this low accessibility of the client stakeholder(s), summarised in table 4.4 below. Interestingly, it can be observed that there seems no common factor that triggers the occurrence of this particular barrier across the participating organisations, which indicated the variety of circumstances faced by participating organisations.

Barriers	Factors	Number of organisation
Difficult access to key stakeholders	*Different geographical location*	**3**
	Busy or indifferent people	**2**
	Complexity of knowledge	**2**
	Cost	**1**
	Delegation of responsibility	**1**
	Multiple layer of stakeholders	**1**
	Office policy	**1**

Table 4.4 – Factors influence the occurrence of difficult access to key stakeholders

For several of the participants, the difficulty in accessing the client is due to the different geographical locations of the two groups. This situation hindered them establishing face-to-face communication and providing quick feedback, the importance of which was explicitly emphasised by organisation 7 and 10. According to organisation 1's participant, the presence of geographical distance means the communication and knowledge sharing is more expensive for them. Thus, the organisation rarely established a communication channel (e.g. focus group) with foreign clients.

For few of the interviewees, the difficulties in accessing the key stakeholders are because the clients were too busy. One of the interviewees felt that the client might be just not interested in engaging in developing shared understanding. Organisation 4 described the problematic situation where lack of access led to a delegation of responsibility to a less senior client stakeholder, who did not have the authority to make certain decisions required by the vendor. Therefore, getting commitment from high-level management as soon as the project starts seems necessary to avoid any potential problems caused by low accessibility.

According to one of the participants, complexity of the domain knowledge could be an obstacle in accessing key stakeholders, as more complex knowledge means a higher level of expertise is required. In some cases, the requirements could reach the level of specialist, thus creating difficulties for the vendor to find someone with adequate expertise to support the interpretation of the shared knowledge. Saeidian and Dale's study (2000) identify complexity of knowledge as a problem in where the clients themselves may find difficulty in understanding the knowledge. Unfortunately, it seems that there is no other way to handle this kind of barrier besides employing a domain expert or develop the vendor's own expertise as the project progresses.

Another factor that can lead to difficulty in accessing the key stakeholder is multiple layers of stakeholders. This particular factor refers to a situation where a client representative met by the vendor is an agent acting on behalf of the actual client. This certainly affects the development of shared understanding, as the layer of interpretation is thicker compared to the situation where the vendor deals with the real client. Moreover, this multiple layer of communication also certainly affects the communication of feedback on vendor's understanding.

Lastly, internal office policy can also be a factor that prevents the vendor from having direct access to the client. For one of the participating organisations, their current office policy does not allow the development team to have a communication with the end-user of the product. As a result it prevents them from clarifying their understanding of the problem. For them, the only chance to get feedback from the users is after the product is launched to the users.

C. The time dimension of communication

Another communication-related barrier identified by the participants relates to the frequency and timeliness of communication activities. Participants identified that they were sometimes not able to communicate with clients in order to refine or verify understanding, particularly face-to-face, as often as they wanted to or at the time of the need. They went on to say that this would often result in them having to make assumptions, decisions or interpretations that had not been verified with the client and so were higher risk (and too often were incorrect). Reasons given for the occurrence of these situations of less than optimal frequency, tardiness or overly brief communications interactions included a lack of client accessibility (and its root causes discussed previously); lack of client commitment or client indifference to the project; overly committed client (too busy with work outside the project); unrealistic project timeframes; and project processes or methodologies that did not put much attention on communication with the client.

D. Mode of Communication

Interviewees identified a number of modes of communication used during project development including synchronous modes (face-to-face, telephone, web-based video conference) and asynchronous modes (email, voicemail). The interviewees generally agreed on the richness of face-to-face communications, in line with findings in literature (Al-Rawas & Easterbrook, 1996; Cao & Ramesh, 2008; Damian, 2007; Zowghi et al., 2001), and all agree that it was preferable to meet clients face-to-face at key communication milestones. They saw the lack of face-to-face meetings as a barrier to sharing understanding. In addition a number of participants specified the importance of key members of the development team being able to call or email clients directly when the communication need was urgent in time but not too significant in impact. It was considered a significant barrier to sharing understanding in a timely fashion if the appropriate developers were unwilling or unable to do this. One organisation sends the key developers on a project to the client's site for a face-to-face meeting near the start of a project. They claim that this improves the subsequent communications of the developers with the clients by getting them "closer" to the client and their environment, improving understanding, trust and empathy.

4.3.1.3. Representation related barriers

During the communication process between the vendor group and the client group, representations of knowledge or current shared understanding are usually utilized to support the development of further shared understanding between the two groups. Text, graphical models, prototypes, and the software product itself were all identified as modes of representation of understanding by participants. However, based on the participants' experience, they claimed that utilization of certain representations can hamper the development of shared understanding.

As summarised in table 4.2, most of the interviewees identified that the use of an unfamiliar or complex representation can be a significant barrier to sharing understanding because the client (or some in the vendor group) don't know how to interpret the representation. Most participants also noted that the inherent ambiguity of natural language could also interfere with sharing understanding where the vendors and clients interpret words or concepts differently. Three participants identified that dry or voluminous representations tended to be difficult to get engaged with and described these "unattractive representations" as a barrier to sharing understanding because they were de-motivating.

All of these barriers are discussed in more detail in the following sub-sections.

A. Unfamiliar or difficult representation

The notion of unfamiliar or difficult representations was strongly identified by interviewees as a significant contributor to representation-related barriers. Interviewees described this barrier as concerning the obstruction of shared domain understanding development due to the use of knowledge representations which are unfamiliar or difficult for the stakeholders to comprehend. In some respects it is similar to the use of jargon, in that it represents a "language" that does not have a common meaning (or any meaning in some cases). One of the major differences is the effort required to develop the common "language". If it is desirable to develop a shared understanding of jargon because, for example, the precision of the jargon is needed to usefully describe the application domain concepts (as opposed to replacing the jargon with more familiar language), this is generally less effort than trying to develop shared understanding of a complex graphical representation such as, for example, a UML sequence diagram.

Interviewees identified a number of notations they made use of to represent domain knowledge, including: flow charts, business process diagrams, scenarios, use-cases, requirements specifications documents, UML diagram. Representations of aspects of the solution space such as screen shots, prototypes, or product demonstrations, were also considered as representations of shared understanding of the problem domain. The notion is that they contain “embedded” understanding of the problem domain and that use of them often uncovered misunderstandings about the problem domain.

Semi-structured and unstructured natural language were identified as the most commonly used representations of understanding. When probed, the reason given for this was that this was viewed as being generally understandable by clients for confirmation of shared understanding.

These linkages between this barrier and some representations (e.g. UML/object-oriented representation and use cases) have been identified in recent literature, for instance see Lubars et al. (1993); Emam and Madhavji (1995); and Karlsson et al. (2007).

Analysis of the transcript reveals that the most common scenario described for the emergence of this barrier is the vendor’s use of representations unfamiliar to the client group. This finding suggests that the initiative for overcoming or preventing this barrier should start from the vendor group themselves.

One of the participants indicated the importance of being aware of the client’s skills and experiences in dealing with certain types of representation, as it is indicated in the following quote:

> *‘I think you need to be careful how you actually use them (representations like UML). It depends on the client. It depends on what style they read as well. I mean we have some clients, commercial clients that they don’t even know how to use*

> *the computer … … … you got to start right back to basic …' –* ***Organisation 10's Director***

The above quote also implies that there is a need for the vendor group to communicate with the client in order to develop such awareness on the client's skills and experiences. Later, this awareness of client's skills and experiences can be used for determining type of representation used for communicating the vendor group's understanding of the problem domain.

Another participant suggested that rather than always conforming to some standardised representation of understanding, it is important to be flexibility and use representations that are easy for the client to understand or learn. The type of representation suggested by Organisation 6's Project Manager is something that she regarded as 'pictorial' (e.g. screen shots, prototype), which is in agreement with findings from studies such as Lubars et al. (1993); Emam & Madhavji (1995); and Karlsson et al. (2007).

Interestingly, there is no suggestion from the interviewed vendors that they could provide training for the client to understand difficult or unfamiliar representations. Perhaps, this is due to the time and cost required to provide such trainings, since time and cost were often mentioned as very constraining by many of the participants in this study.

B. Ambiguity of natural language

Four participant interviewees identified that ambiguity of natural language can interfere with developing shared understanding in the sense that different understandings could be developed from one source of (ambiguous) information. This can obviously result in unsatisfactory solutions and interviewees stressed the need to have mechanisms to identify ambiguities in documents such as requirements specifications, as exemplified in the following quote:

> *…they (client) use the word that is normal in common life, but for them it means a very specific thing and you've got to be able to pick up on that and ask for an explanation or confirm the real meaning.*

Identifying and clarifying ambiguity typically involved document reviews and frequent (iterative) confirmation of interpretations by verifying against other representations, such as diagrams and prototypes, or sometimes more formal structured representations (e.g. UML, Use Cases). In many cases interviewees described the use of a quick phone call or email from the vendor stakeholder (e.g. developer) to the appropriate client stakeholder to clarify ambiguity.

Another participant claims that “written communication in any form is ambiguous” and regular face-to-face meetings will reduce this ambiguity, so and “regular contact is the key…so no matter what we write, we try to get together with the client as often as we can…”

A similar suggestion is also indicated by Karlsson et al. (2007) who identify that the utilization of regular communication in form of group discussion for increases the clarity and understanding of the requirements between the vendor and client. This adds further credence to having regular communication between the vendor and the client groups in the context of developing shared understanding between those two groups.

Many studies describe the issue of ambiguity, particularly in relation to requirements specification (se for example: Gervasi and Zowghi (2005); and Karlsson et al. (2007)).

The importance of being able to detect the natural language ambiguity is in agreement with Berry and Kamsties (2003), who suggest the importance of learning to detect ambiguity and imprecision of natural language representation.

C. Non-engaging representation

A few of the participants addressed their concern on the “unattractiveness” of certain representations used in communicating knowledge. They believed that this could affect the interest of the other parties in engaging with such representations, causing boredom and lack of motivation to review them. Typically “unattractiveness” referred to dry, plain, dull, voluminous (“thick”) documents for review or confirmation of understanding.

In the literature, a similar situation has been discussed as well by Cederling et al. (2000) concerning the ineffectiveness of using thick paper-based documentation in developing shared understanding, due to its “impenetrability”.

One organisation, for which paper based documentation is considered the main mode of communication (e.g. organisation 8), divide the documentations into smaller chunks for regular meetings, avoiding overwhelming clients with volumes. One organisation suggested that guidelines for matching a document’s “format” with a specific project and client in order to “optimise” the review process and refining shared understanding of a problem domain could be a potential fruitful area of research.

4.3.2. Stakeholders involved

One focus of this research is a deeper understanding of the nature of the stakeholders involved in the process of developing shared domain understanding. The characteristics of the stakeholders could be a significant factor in designing better processes or tools for developing shared understanding and knowledge sharing. Information regarding the stakeholders involved is mainly gathered through questions 6 to 9 of the interview questions (Appendix A).

Analysis of the transcripts of the interviews indicates that, in general, the activities of gathering domain knowledge from the client group, analyzing that knowledge, and thus developing a deeper understanding of the problem domain, are driven by the vendor group. Sharing this understanding with the rest of development team is usually regarded as the responsibility of the vendor also. In a related study, Aranda et al. (2007a) refer to the role responsible for these activities as the ‘*analyst role*’.

In this study, however, only three of the organisations identify the analyst role explicitly. In another three of the organisations the analyst role appears to be taken by the head of the organisation or business owner. The other four organisations assign the role of analyst to senior figures like project managers or consultants. There were no explicit explanations given regarding the appointment of personnel to do the analyst role, however, some possibilities may be inferred. For example, in one organisation, each member of the organization, including the CEO herself, is expected to engage with the client (e.g. in terms of support, help, developing proposal), and this may explain the "diffuse" nature of this analyst role in that organisation. For other organisations, skills essential to the analyst role (for instance communication skills or negotiating skills) were perceived to be present in only senior staff. For some organisations the organization's leader or founder "controlled" this analyst role. The situations where the CEO, business owner, or other senior figures take the role of analyst supports related findings from Aranda et al.'s (2007a), in their recent study of RE practice in small to medium software organisations.

People in the analyst role can generally be categorized in the mediator role also, since they apparently become a "connector" between the client group and the rest of the development team.

Apart from analyst role, several other roles, which are relevant to development of shared understanding, are also identified from the interview result. They are:

- ***Interpreter:*** this role supports the knowledge transfer process between the vendor and the client group by helping the interpretation of certain knowledge which is difficult to understand without the availability of domain expertise. In this study, this role is found within organization 1 and 5's System Consultant and Organization 5's Technical Director who have significant domain expertise. According to Lubars et al. (1993), this particular role is the role that is usually assigned to employed domain experts. This role can be considered as a mediator role as well, since people with this role support the communication between the client and vendor groups by bridging the communication of the two groups.
- ***Supervisor:*** this role involves supervising the progression of a project, including the development of shared understanding and ensuring that the communication between the vendor and the client group representatives is occurring and within the scope. This role is found within organisation 3's IT Delivery Manager, organisation 9's Solution Manager and organisation 10's Director. This role

appears to be very similar to the facilitator role found in literature (Coughlan et al., 2003).

- ***Regulator:*** In this study, the existence of this particular role is identified in organisation 8’s project. The regulator was a representative from a government organisation whose role is to provide information regarding legal compliance of the implemented product. In this case, the rules are absolute and could not be negotiated. The involvement of a regulator is also indicated by an organisation which publishes standards related to museum data. However, for most of the participating organisations, the involvement of regulator can be considered an unusual case.
- ***Decision maker:*** Even although it is not explicitly stated by the majority of the participants, people with decision maker role during the process of developing shared understanding can be inferred from the transcript.

> *‘…the main problem is when we talk about there’s not one person who can make the decision.’* ***– Organisation 4’s Owner***

Many of the participating organizations (organization 1, 3, 4, 6, 9, and 10) adopt the practice of encouraging the direct involvement of developers (vendor group) with the client stakeholder group. A better understanding of the clients’ points of view and a clearer understanding of the client’s expectations are reported, by organisations 3 and 1 respectively, as being the main benefits of this practice. Moreover participants from organisation 3 also reported that this “unrestricted” communication resulted in less need for the developers to provide written artefacts to share ideas with the clients and measurably allowed the developers more time to spend improving the quality of the software.

On the other hand, some organisations don’t use this practice (e.g. organisation 2, 5, 7, and 8) and discourage “free” communication between the clients and software developers. This was sometimes as a result of the vendor organisations’ policy or the business owner’s preference, and was generally a result of previous bad experiences. One of the participants (organisation 5) indicated that they wanted to “shield” their developers from the complexity and diversity of the clients’ actual configuration and implementation processes. They used a domain expert to liaise with the client and developers instead. For organisation 8 and 10 the prevalent view was that ‘developers are merely technical’ and this was reason enough to not get them directly involved with client stakeholders during the development of shared understanding on the problem domain. This was based on experiences such as described by organisation 10:

> *‘…we had some projects failed badly, because the developer couldn’t comprehend the business process at all and the developer was approaching from a very technical angle…’* – ***Organisation 10’s Director***

Organisation 10 suggested developers could be involved in discussions with the client when the discussion has entered the product stage (e.g. workflows of the product).

Although many of the participating organisations adopt the practice of involving developers in developing shared understanding with the client, there is little explicit evidence from the interviews to suggest that the interviewees perceive this practice as important for enhancing the developer’s understanding of the problem domain. The view of the developers regarding the need for communication with the client to develop shared domain understanding was not represented in this study because none of the interviewees was a developer. This would be an interesting extension to this research. Similarly the view of the client side stakeholders isn’t represented in this study but is planned as a follow-up study.

The participants stated that the client group is typically represented by management–level stakeholders (e.g. CEO, CFO, Head of department, managers), representatives from the client’s Information Technology (IT) department, a Business Analyst, and users (e.g. expert users and end-users). Interviewees identified that, the client stakeholder group is often responsible for providing the vendor with much of the domain knowledge needed to do the project (e.g. process related knowledge), although some vendors contribute their own domain knowledge. In addition the client representative is seen as have the role of validating and providing feedbacks on the vendor’s current understanding, and making decisions about alternative ideas when it is needed. In some cases, the client representatives take a role as communication liaison to the other stakeholders from the client side (which are typically the end-users). As for the users, they are typically expected to share their operational knowledge on how they would like to (expect to) use the software, as well as to validate the functional aspect of the product (e.g. through acceptance testing). Future research on the client-side perceptions will reveal whether the clients' perceptions of their roles align with the vendors' perceptions of the clients' roles.

From the interview data, then, the vendors seemed to have a fairly restrictive view of the stakeholders' roles and their relationships, as summarised in figure 4.1.

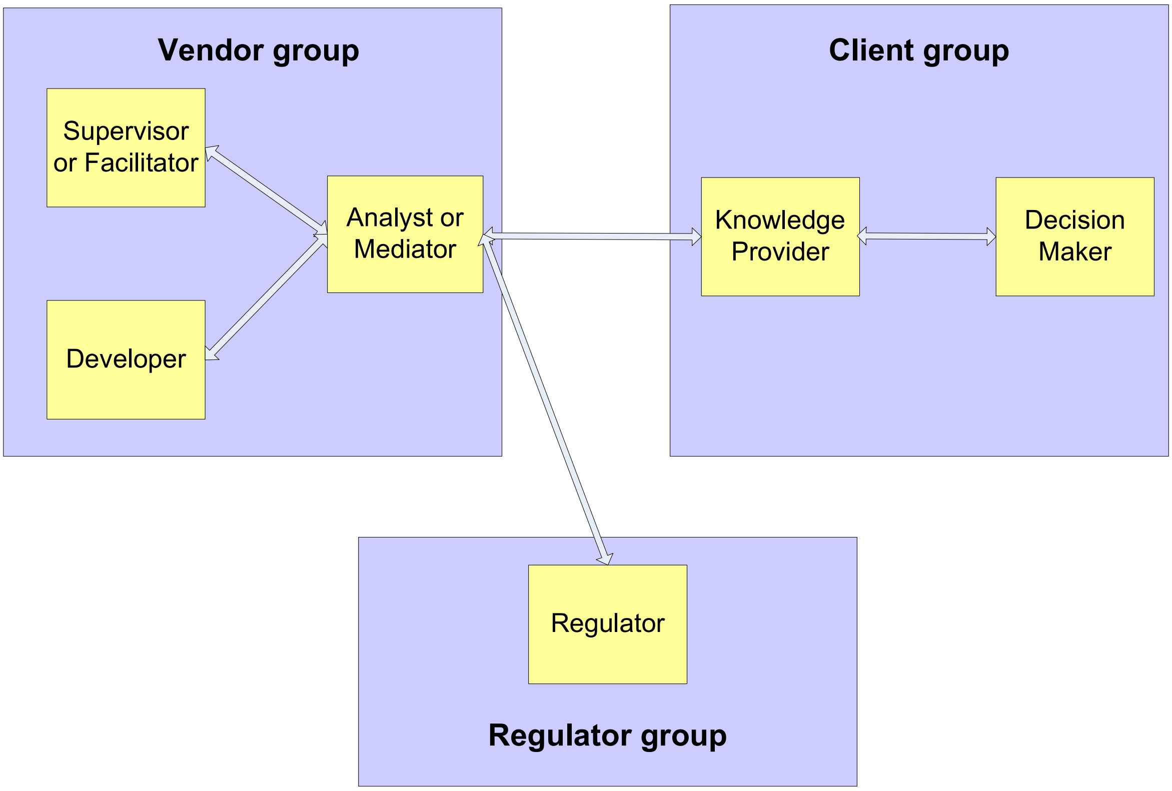

Figure 4.1 – Relationship among involving stakeholders' roles

4.3.3. Process of developing shared understanding

The second research question of this study investigates what practitioners (vendor group) say they *do* in order to develop shared understanding of the problem domain with the client stakeholders. Information relating to this matter is gathered mainly from questions 8 to 11 of the interview questions (Appendix A), which ask about activities done to obtain and confirm their understanding with their clients (or potential clients). The motivation behind this is to identify and analyse any research-practice gap (i.e. differences between what can or should be done according to literature and what is done in practice). Also a long term aim is to improve current practice, and this presupposes an understanding of current practice, which this part of the study starts to investigate.

In general, when asked to describe their current process (for developing understanding of the domain problem or opportunity), all of the participants refer to the RE process used by their respective organisations. This suggest that the vendors perceive a strong relationship between RE activities and developing shared understanding of the problem domain, supporting a similar view found in literature (see Chapter 2).

A typical interviewee's description of the process of developing shared understanding in terms of phases of activities is:

- Initial elicitation
- Knowledge analysis
- Artefact development
- Validation and verification
- Finalisation

The following sub-sections describe and compare the interviewees' perceptions of these phases in more detail and discuss some interesting conclusions and their implications. Also described are the participants' descriptions of tools and techniques they use to support activities that develop shared understanding, as well as their judgement on their efficacy

A. Initial elicitation phase

Unsurprisingly, there is a clear distinction between the elicitation activities described by participating organisations which are software service companies (involved in bespoke software development) and organisations which are product-driven.

For the participating software service companies the process of developing shared understanding usually starts during the initial meetings or contacts with the client group's representatives. For most of the participating organisation, these meetings are usually done in face to face situation. However in some cases, where physical interaction is not possible (e.g. due to vast geographical location (organization 4)), this knowledge sharing process can be done through e-mail, phone, or web-based video conference meeting.

During this initial phase, the knowledge being shared usually comprise of goals, scope of the project, idea, concept, and so on. Furthermore, it is also possible for the client to provide their own requirements specifications or documentations straight away, as it is experienced by some of the participants (organisation 4, 6, and 8).

In terms of technique used to elicit the knowledge, techniques such as interview with key stakeholders, observation on current process, or conversation which consist of questions-answers are usually used in this particular phase.

As is indicated by a few of the participants, it is possible for this particular phase to consist of more than one meeting or session, especially if the project's level of complexity considered high (e.g. dealing with stakeholders from different organisations apart from the client group, for instance: government organisations).

On the other hand, for participating organisations which provide market-driven software products (organisation 1, 5 and 6), activities they described as elicitation of the domain knowledge include facilitating focus group meetings with representatives from user groups, and collation of entries from current or potential users via issue tracking systems. As Lubars notes (1993) this difference in domain knowledge elicitation between software service and software product companies may, at least in part, be attributed to the much wider user base of software products compared to bespoke software. This may result in the vendors of a product being "further away" from their market and user base because they don't deal directly with the diversity of end-users, either because they are geographically separated or the supply channel involves intermediary distributors.

Interestingly, several of the participants appear to use the notion of the development of shared domain understanding and the development of requirements (user and software) interchangeably when describing elicitation activities. They often refer to the "business problem", "the problem" or "the requirements" as the object of their shared understanding. The prevalent view of the participants appears to be of the vendors "extracting" enough understanding of the problem domain from the clients to start developing a candidate software solution, and less emphasis on it being a collaborative phase, in general. Certainly the impression gained from the interviews is that the vendors generally are the drivers and instigators of knowledge elicitation activities.

Analysis and integration of this elicited domain knowledge into their existing knowledge is described in the next section.

B. Knowledge analysis and artefact development phase

Most of the participating organisations also describe a phase of more intensive knowledge analysis where they are analysing processes, process dependencies and other aspects of the application domain that they have elicited. These activities were often intertwined with knowledge elicitation and clarification activities with client stakeholders as the vendors realised what they didn't know (or sometimes what the clients didn't know).

Two of the participating organisations specifically noted that, if the project domain is unfamiliar, their analysts would conduct further research on the application domain by seeking knowledge from colleagues, the internet, or some other source external to the client/vendor stakeholder groups. This domain knowledge from external sources may or may not be shared with the client group. Sometimes it was part of developing an image of appearing as domain experts to the clients so that they enhanced the client's confidence in them.

All of the participants indicated that they develop some artefacts as part of this domain knowledge analysis phase either to support analysis or (and) to represent the new domain understanding for future sharing. Examples of such written artefacts commonly identified by the participants are business process diagrams, project proposals, work flow diagrams, scenarios, specification documents and functional specifications. Other artefacts described by the interviewees as resulting from the knowledge analysis are software artefacts such as prototypes or parts of an existing product. For few of the participating organisations development of prototype or parts of the final product appear to start after at least one cycle of the project or when some fundamental requirements haven been signed off or agreed to. However, for three of the participating organisations, they describe the desire for earlier development of prototypes for reasons specific to their application domain. For instance, organisation 2 is involved with the integration of specialised mechanical hardware and software, and the interviewee claimed that presenting a prototype (how the machine works with the prototype software operating it) is the most effective way to represent and share their current state of understanding of the client's needs, (which they equated with domain understanding.

Often these artefacts developed as part of the domain analysis are then used to verify aspects of the vendor's understanding of the problem domain with the client. Activities and insights relating to this phase are described in the next section.

C. Verification and validation phase

All participants described a phase of confirmation of their own understanding of the problem domain with the clients, usually iteratively with the previous phases, and generally involving sharing one or more of the artefacts previously described. Verification activities identified by the participants include walking through and explaining the artefacts (documents or software) face-to-face with the clients, as well as sending artefacts (more often documents) to the appropriate client stakeholders for them to read, review and send comments back to the vendors. There may be very little delay between domain analysis and verification activities, for example at a kick-off meeting with clients and vendors, or longer delay where elicitation and analysis are quite complex and protracted. All participating vendors agreed that the quicker there was an artefact to discuss and get feedback from, the better.

A few participating organisations distinguished validation activities from verification ones and described them as activities where appropriate client stakeholders "test" an artefact developed by the vendor group.

All participating organisations described the verification phase as including activities such as meetings, workshops, or video-conference meetings, all of which involve synchronous face-to-face (physically or electronically) communication between the vendor and the client groups. All interviewees identified email and phone calls as reasonably common in confirmation or clarification activities also. Electronic communications was (unsurprisingly) prevalent in geographically separated client/vendor groups, with one participant making heavy use of internet-based video conferencing. Four participating organisations indicated their preference for doing this kind of activity in a face-to-face mode, claiming that there were clear benefits to understanding by being able to have immediate responses, active participations from the client group, see body language, as well as providing more opportunity to uncover other problems and opportunities. Other participants highlighted the importance of less formal and ad hoc verification of simple concepts via phone calls or email.

For most of the participating organisations, the process of verification and validation is done in an iterative manner. This iterative process usually involves (a) meetings (or other communication with the clients) to obtain thee feedback on some artefact(s) that represents the current state of understanding about some aspect of the problem domain, (b) reflection and integration of the feedback with the current state of understanding (separate from the client group), and then (c) further communication with the client group to confirm this new understanding and negotiate a shared understanding. This is in line with current literature which demonstrates that the iterative nature of these processes is an important aspect of shared understanding development (e.g. Damian, 2007; Desphande et al. (2005)).

The frequency or regularity of this iterative elicit-analyse-verify process is varied among the participants appears to be influenced by factors such as project deadline, availability of client and vendor representatives, and the complexity of the problem domain. Several of the participants reported that the frequency of meeting is usually discussed during the initial meeting of a development project.

For the product driven organisations interviewed, which may be interested in verifying understanding with a “market”, the process of verification and validation is mainly done using representative domain experts selected by the vendors for this purpose. For instance Organisation 1 found this expert at a focus groups meeting with representatives from different user groups. Organisation 5 found this domain expert at an existing client installation involved in the current release cycle. These experts seem to have a role similar to the client representative of bespoke type organisations in terms of providing feedbacks on the artefact developed by the vendor groups.

D. Finalization phase

Following the agreement on a refinement of shared understanding, the participants describe a finalization phase where the vendor group implements any changes that are a consequence of this latest evolution of domain understanding. This finalization process is typically done when there are no further comments or questions raised by the client group regarding the artefact, by “signing” a document (e.g. requirements specification). This was perceived by six of the interviewees as a sign that shared understanding of this aspect of the problem domain had been “achieved”. However, in several organisations, the “sign-off” process is less explicit (verbal or informal email) and appears to be more reliant on the trust built from collaboration on previous projects.

For the participating product driven companies, where generally product development is not dependent on individual clients, the finalization phase may involve:

- Independent domain experts’ feedback
- Information stored in issue tracking system (organisation 1 and 5) or support department (organisation 6).
- Recommendations from the vendor’s sales, marketing or support staff who regularly interact with existing or potential clients.
- Report documents from user groups meetings in different areas.

- Other documentations published by other organisations, for instance: documents about standards or existing clients' white papers.

The interviewees describe how the information from these sources is synthesised and aggregated, and prioritised as candidate changes to be implemented in the next product release. For one organisation this elicitation, analysis, verification and finalization of refinements to the domain understanding is the responsibility of the vendor's Business Analyst group solely The developers and systems analysts rely on the ability of the business analysts to evolve the domain understanding with the client stakeholders and share it with the development team. The interviewee pointed out the increased risk of an interpretation or decision that does not reflect the development team's current shared understanding of the domain. This was a political decision based on some previous poor experiences of a developer/analyst interacting negatively with a client representative.

The next section describes the tools and techniques that participants identify as supporting these activities involved in sharing understanding.

4.3.4. Tools, techniques and their efficacy

One of the dimensions of interest under the topic of shared understanding development is the tools and techniques employed to support the development of shared understanding between the vendor and the client groups. Therefore, some of the interview questions are designed to elicit this kind of information from the participants (see questions 12-14 in Appendix A). Investigation on this particular area is intended to inform the readers about the existing tools and techniques that are considered helpful by practitioners for supporting the development of shared understanding, as well as to provide a base for developing new ideas on tools, techniques for future researches on this particular dimension. A summary of the tools and techniques identified by participants is presented in table 4.5 and discussed in more detail in the subsequent sub-sections.

Tools or techniques	Type	Typical functions	Usual time of used
conversational techniques (e.g. conversation, interview)	*Te.*	*eliciting & verifying knowledge*	*elicitation; verification & validation phase*
white boarding session	*To. & Te.*	*triggering idea & eliciting knowledge*	*elicitation phase; verification & validation phase*
focus group	*Te.*	*eliciting knowledge*	*elicitation phase*
Questionnaire	*Te.*	*eliciting knowledge*	*elicitation phase*
Observation	*Te.*	*eliciting knowledge*	*elicitation phase*
Similar or working products	*To.*	*triggering idea & discussion*	*elicitation phase*
Artefact development tools (e.g. Visio, PowerPoint)	*To.*	*developing artefacts*	*analysis & artefact development phase*
document + walkthrough (or review)	*To. & Te.*	*validating knowledge*	*verification & validation phase*
prototype + walkthrough (or review)	*To. & Te.*	*validating knowledge*	*verification & validation phase*
User acceptance test	*Te.*	*validating knowledge*	*verification & validation; testing phase*
Issue tracking system	*To.*	*knowledge storage & management*	*whenever required*
Glossary	*To.*	*providing information on difficult terms*	*whenever required*

Table 4.5 – Tools and techniques used in developing shared understanding

(Note: To. = Tool; Te. = Technique)

4.3.4.1. Conversational techniques

Conversational techniques, which range from casual conversations to formal interviews, (both face-to-face and by telephone), are identified by participants as the most commonly used techniques. This appears to be because of their wide applicability to all aspects of domain understanding and communication between the vendor and the client groups. Generally, these techniques are used by the vendors in every phase of developing shared understanding. Moreover, conversational technique, particularly verbal discussion, is considered as the main technique for handling disagreement in understanding by the majority of the participating organisations.

Some of the participants indicated the need for people with good communication skill in order to execute this technique well, as the better the communication skill, the more effective the execution of knowledge sharing activities. They stated that the selection of stakeholders with good communication skill is crucial for developing shared understanding. This supports studies in literature (e.g. Curtis et al., 1988; Emam & Madhavji, 1995) which demonstrate the importance of good conversational communications skills as a factor in developing shared understanding. In relation to this matter, a significant effort is shown by one of the participating organisations (organisation 9) which invests some of their financial resources for improving their staff's communication skills by providing regular trainings from a communication expert.

4.3.4.2. Focus group

As described by Nuseibeh and Easterbrook (2000), a focus group is one of the group-based techniques for eliciting requirements and to some extent it can be used as well for assessing vendor's current understanding of the clients' problem domain.

In this research, it is found that the use of focus groups is popular among product-driven organisations. A focus group's strong point is its ability to facilitate wider participations from the client group to engage in discussion on a focused topic lead by a neutral moderator. Thus, it is beneficial to the vendor for gathering rich information and beneficial to the client group (participants) to learn from other participants as well and update their current state of domain knowledge.

One participant asserted that focus groups were very effective for validating their current version of their software product, by providing quick feedback from a diverse range of potential users.

Furthermore, the same organisation also sees the focus group as an opportunity to look for a credible domain expert whom they can ask for advice regarding their product. This could be considered a significant benefit for product driven organisations, since they tend not to have fixed client that can provide requirements and feedback authoritatively (Lubars et al., 1993).

4.3.4.3. Document+ walkthrough

In the context of developing shared understanding, a document can be considered as a tool, since it can be used as a communication device between the vendor group and the client group, as noted by some of the participants:

> *'…giving people something to communicate around … … … some template [document] that people can become comfortable with not in the way that we change things, but because we have something to talk to.'* ***- Organisation 5's Technical Director***

> *'So that (document) is our main communication device for most of our project.' –* ***Organisation 6's Project Manager***

A document, in the context of this study, typically is a written specification document or requirements document, with content that can be text, diagram models, etc.

However, participants noted that a document on its own is often found to be ineffective in developing shared understanding due to problems like ambiguity or inconsistency of the document content as exemplified by the following excerpts from the transcripts:

> *'…develop some kind of documentation … … … But these are very dry documents. They do not near help people in reality…. … … You never know, how much attention people have actually had when they were reading the documents, what is behind their sign-off, what kind of understanding … you know words are only words…'* ***– Organisation 3's IT Delivery Manager***

> *'…written communication in any form is always ambiguous, and so no matter what we write, we try to get together with the client as often as we can.'* ***– Organisation 9's Solution Manager***

This supports one of the findings in a recent empirical study on communication practice in RE by Segal (2005).

Further discussion regarding this ineffectiveness of just documents with no face-to-face discussion has been addressed in the previous section (4.3.1.3). Concerning the ineffectiveness of solely a document, as is implied in organisation 9's comment previously quoted, even if the understanding has been documented, communication with the client group is still important to ensure the accuracy of understanding developed by the vendor group. In relation to communication, the use of a document in developing shared understanding is often found to be combined with a document walkthrough or document review. This allows the vendor to go through the content of the document to see and discuss whether there are problems (e.g. inconsistency) related to the recorded knowledge, potential risks identified, or misunderstandings between the vendor and the client stakeholders.

The participants interviewed agreed that a document review is more effective if it is executed in a face-to-face situation, since it ensures participation of the client group. Face-to-face artefact reviews are suggested as best practice by Hofmann and Lehner (2001). In addition, participants identified that a document review can also be done asynchronously by the client stakeholder group, although they noted that this can be less effective since a client may be unwilling to spend time reading the document provided or at least won't have the "immediacy" pressure of the face-to-face review.

Another obstacle to the use of documents for enhancing shared understanding that participants identified is the sheer volume of the documents and the resulting time and effort required to process these documents. In this case, partitioning the review process into a number of sessions based on document's sections, topics or chapters is recommended, as addressed by one of the participants in the following quote:

> *'…usually our requirements specification … … … will get to between 90 to 100 pages … … … So each meeting, you may be working on 3 or 4 pages or possibly if you're at higher level, you may cover 2 chapters at the maximum, trying to get very focused. That's what I do, because it works much better.'* ***- Organisation 8's Director***

Interestingly, even though some of the participants are found not to rely on documentation in developing shared understanding, a signed-off document is still considered important as a proof of agreement due to commercial reason, as typified in the following quote:

> *'…for our customers' projects, we have to have documents, which is what get signed off in form of contracts, we have to have it.'* – ***Organisation 6's Project Manager***

4.3.4.4. Prototype + walkthrough

A prototype is considered as a tool for developing shared understanding as well by the participants. A prototype acts as an operational model, which can be experienced by the client group (Hofmann and Lehner, 2001). In this study, the majority of the participants rely on a prototype or at least mentioned the use of prototype when asked about activities that relate to the development of shared understanding.

The participants identified a number of interpretations of the concept of a prototype including:

- **Paper prototype.** This is described by one organisation as paper-based screen mock-ups used to obtain better understanding on how users would use the product.
- **Web based prototype:** This prototype is basically screen-shots placed in a web environment to be accessed by clients in different geographical areas. The screen shots were combined with a textual description, and an on-line feedback form. The interviewee considered this type of prototype as useful for gathering information from wider users.
- **Screen shot based workflow:** This variant of prototype combines screen shots from the client's existing system to animate the workflow of an existing product offered by one of the participating organisations.
- **Executable model:** This type of prototype represents something similar to a flowchart or a diagrammatic type model. However, the model can receive input from the stakeholders, executed to demonstrate the data flow and automation of transactions occurring in the business process depicted by the model.

A prototype is usually created by the development team from the vendor group and brought to a meeting, workshop, or focus group to be presented to the client stakeholders. It is discussed in terms of its impact on the problem domain, analysed, and validated against the requirements. From the discussion and the review and validation process, feedback from the client group is obtained and used by the vendor group to refine the prototype. All of these processes are usually done in an iterative manner until an agreement is reached between the two groups. These practices are corresponding with Hofmann and Lehner's suggestion of best practice in RE (2001).

In terms of effectiveness, participants generally described prototyping as very effective for developing shared domain understanding because of its visual and tangible characteristic, which allow stakeholders to see, test, experienced the prototype to some extent. According to one of the participants, these characteristics are especially effective with clients who are weak at conceptualising part of the knowledge, which tend to be more abstract, as evidenced in the following quote:

> *'Very few people can assess a concept, they have to see it manufactured…'* ***– Organisation 2's Owne****r*

In addition, due to its visual characteristic, a prototype is also considered useful for bridging the communication gap caused by differences in first language, as indicated in the following comment:

> *'…because they're dealing with Japan they have used Power Point a lot and put a lot a pictures because you can actually mimic a work flow (prototype)… … … dealing with people who have English as their second language you don't want this fancy, very wordy stuff …'* ***– Organisation 6's Project Manager***

However, according to one of the participants (organisation 3), practitioners need to be cautious in employing a prototype, because it may result in unrealistic expectations. This particular situation is clearly illustrated from the story from participant organisation 10:

> *'We had a disastrous one recently where we showed someone a website – it gave her some ideas about what you could achieve. And it was terrible, because what happened after that is that she gave us the requirements and we started to*

> *develop them and then she said: "Oh.. I've been to this other website and it's got this other things, and I'd like to have …" and what happen after that was that she was constantly going to other websites and adding something to her requirements….'* – ***Organisation 10's Director***

The situation addressed in the story above is consistent with a finding from past study by Cao & Ramesh (2008) who investigate the agile requirements engineering practice. Consequently, such an experience suggests that prototype is best to be employed if the client has already had certain level of details which are reflecting what they really want to achieve. On the other hand, if the client is not ready, perhaps a consultation or a white boarding-session would be appropriate to facilitate the construction and understanding of the client's idea.

4.3.4.5. White-boarding or sketching

White-boarding or sketching is often identified by the participants in relation to developing shared understanding. The terms refer to a technique where both vendor and client representative are collaborating on a same physical collaboration space (which tend to be a white board or a piece of paper) to establish a shared understanding on an idea, a concept, workflow, a design, etc.

According to one of the participants, the effectiveness of this technique is located in its ability to encourage contribution of the client group as well as allowing immediate feedback from the client group through combination of graphical and verbal representation. Similarly, Dekel and Herbsleb (2007) suggest that this particular technique is effective because it allows the stakeholders to focus on the problem, as well as encouraging experimentation with the design.

In terms of timing, generally, this technique is often used during the early design phase of the project, which is consistent with Dekel and Herbsleb's observation (2007).

4.3.4.6. Other tools and techniques

The participants also identify some other tools and techniques which are not as commonly identified as those previously described. The techniques are:

- **Questionnaire.** This technique is one of the traditional techniques for eliciting requirements from the clients according to Nuseibeh and Easterbrook's (2000) classification of elicitation techniques. Little information is given about the utilization of this technique, except it has been used by one of the product driven organisation to gain wider information on issues and enhancement request from their foreign clients and confirming the organisation's assumptions on particular issues. The advantage of this technique is in its ability to cover wide range of respondents, which is useful for product-driven organisation in gathering knowledge from wide range of customers or potential customers.
- **Observation.** Only two of the participating organisations indicate the use of observation in their attempt to improve understanding the problem domain. This technique is useful in gathering information on how the clients actually go about their work and for identifying potential issues with these activities. Further enquiries to the relevant vendor stakeholders are required to investigate this technique's efficacy for obtaining information on the problem domain. This particular technique also included in Nuseibeh and Easterbrook's (2000) classification of elicitation techniques**.**
- **User acceptance test.** Similar to the prototype, this technique allows the client to review, validate the product, by conducting a test on the given product. Feedback on the validation process potentially reveals issues such as enhancements, changes, or misunderstandings of the project scope or domain understanding.

Tools less commonly identified by participants include:

- **Issue tracking system and on-line forum.** Both of these tools share a similar purpose, which is providing a channel for the client group (particularly users) to share their feedback to vendor group. This may include enhancement requests, bug reports, feedback on products, and so on.

- **Artefact development tools.** This refers to the tools used by the stakeholders for developing artefacts which are considered useful for developing shared understanding. For instance: MS Visio and MS PowerPoint software for constructing representations such as diagrams and workflows; Rapid development tools for constructing executable prototypes; CAD CAM software for constructing 3D animation model of machinery tools; and flowchart tools for constructing process models.
- **Communication tools.** Apart from e-mail and telephone which have been considered common due to their wide use among the participating organisations, communication tools in this study also include teleconference tool. Teleconference tool is usually used by vendors whose clients are difficult to access physically, as it allows long distance face-to-face communication over screen. Beside that, it also able to display screen shot, thus support the use of prototype based walkthrough technique as well.
- **Glossary.** This tool provides reference for domain specific terms used during the projects and is useful for developing a common language for sharing understanding.
- **Similar or working product.** This kind of artefact is usually used for demonstration purpose in order to trigger further discussion related to product's feedback or design ideas for new products**.** Moreover, it is also useful for explaining workflow related knowledge.

4.3.5. Representation of understanding

Another dimension related to the development of shared understanding in this study focuses on the *representation* of the stakeholders' current understanding of the knowledge that needs to be understood. Participants' responses regarding this matter are mainly gathered from questions 29 to 32 from the interview questions (Appendix A) and they are intended to answer part of the second research question. A summary of the analysis regarding the representations of understanding used by participants can be seen in table 4.6.

Type	Representation	Total
Natural language	*Verbal (e.g. during meetings, phone calls, training sessions, teleconference meetings)*	**10**
	Text (e.g. E-mail, written documents, others)	**10**
Product	*Prototype (e.g. screen mock-up, paper based, executable model or parts)*	**8**
	Products (e.g. similar products, working product)	**4**
	Images (e.g. drawings, screen-shots, animation)	**4**
Graphical	*Diagram (e.g. Process, work flow, system structure diagram)*	***6***
	Flowchart	***4***
	UML type diagram (e.g. Class, architecture type diagram)	***3***

Table 4.6 – Representation of understanding

From table 4.6, it can be observed that the identified representations can be categorized into three major groups, namely: natural language representations, representations related closely to the software product, and graphical representations. The analysis clearly shows that natural language representations (both verbal and text) are the most commonly used representation in the case organisations. The next most commonly described representation of current domain understanding is categorised as product-related representations. These include prototypes, working products, and images of screen shots. Under this category, prototyping appears to be the most widely used among the participating organisations. The third category is graphical representations of understanding, which includes various types of diagrammatic representations such as process diagrams, workflow diagrams, system-structure diagrams, flowchart, and various UML diagrams.

One of the important aspects related to a representation is its degree of formality, because it is related to the degree of accuracy offered by the representation in representing the knowledge (Nuseibeh & Easterbrook, 2000). This in turn can influence the quality of shared understanding developed. In case of this study, the dominance of natural language representation suggests that there seems to be high reliance on informal and loosely structured representations, which is often characteristic of natural language (Gervasi & Zowghi, 2005; Neill & Laplante, 2003). On the other hand, representations with higher degree of formality were generally seen as unfamiliar to the client and so less useful for representing and confirming understanding.

Interestingly, even though formal representation (e.g. Z notation) offers significant advantages, such as: higher level of accuracy in the represented knowledge and automated analysis, (Nuseibeh & Easterbrook, 2000), there seems no indication of its utilization in participants' experiences. This lack of formal representation utilization is probably due to the perceived difficulties in dealing with it (which typically using mathematical notation), since representation with slightly lower degree of formality (e.g. UML) is found to be unpopular due its difficulty among the participating organisations. Other factors like the training, cost, and effort in adopting such representations (Palshikar, 2001) might influence this also.

One of the participants suggests that sometimes it is important not to focus on the existing representation standards, but rather to make sure that the client understands what the vendor wants to convey.

> *'You've go to convey the message. That's the important thing that the receiver needs to understand …'* – ***Organisation 6's Project Manager***

Participants rarely mentioned the persistence of the representation as being an important characteristic of the representation. Several mentioned that often a persistent representation of understanding was not kept current either because there was no desire to do this (it wasn't seen as valuable), or because of poor practices.

Further discussions on each types of representation are addressed in the following sub-sections.

A. Natural language representation

In this study, natural language representation appears to be the dominant representation across the participating organisations. This includes verbal language or conversation, at various interactions such as meetings, phone-calls, training sessions, and teleconference meetings. It also includes forms of written text such as e-mails and named documents (e.g. specification documents). Participants emphasised that the use of natural language is generally complementary to other type of representation, such as: prototype and graphical models.

Focusing on the stakeholders, the use of natural language is found to be common for both of vendor and client group. Generally, in order to develop shared understanding, the client initiates the knowledge sharing process, either verbally in a meeting or by e-mail or phone calls. Typically these will be followed up by the vendor with further discussions and clarifications, mainly through the same mode of communications.

The use natural language representations are generally considered effective for domain knowledge sharing by the interviewees. They reasons they give are that, since it can be used to represent almost every aspect of the domain, due to its "richness" and it generally can be understood by everyone, the learning effort is low and the amount and density of knowledge represented is high. For example, one participant justifies the effectiveness of using a text spreadsheet to share understanding:

> *'…it's because everybody can read it, and it's easy to understand and also it's easy to interpret'* – ***Organisation 2's Owner***

However, as discussed in more detail in section 4.3.1.3 on barriers, the use of natural language is prone to problems related to ambiguity or inconsistency.

B. Product-related representations

Product–related representations refer to representations of domain understanding which to some extent closely resemble the intended software product (solution) being developed. This includes prototypes (e.g. screen mock-ups, paper prototype, and executable models), working products, and images (e.g. drawings, screen-shots, or animations. Prototypes in particular were widely used among the participating organisations and were viewed as important representations of domain understanding that could be "tested" by the client. Even although prototypes are generally associated with the solution domain in literature (Hofmann & Lehner, 2001), participants viewed the use of prototypes as useful for uncovering misunderstandings about the problem domain or triggering “new” views of the problem domain.

For example, a prototype or product to some degree represents how users work as part of a domain process or workflow. Some participants describe how this can potentially lead to identifying insights or better understanding of the domain processes, workflows or users’ work habits. Another case organisation reported the use of screen shots of a proposed system for representing, clarifying and confirming their understanding of certain (domain) business rules with their client.

The use of solution-based representations, such as prototypes, as representations of the *problem* domain is not commonly discussed in literature. Further research into this might be useful, considering the frequency of the stated significance of this type of representation as a representation of shared *problem* domain understanding, in this field study. Perhaps attributes of the (concrete and detailed) product-related representations could be automatically and explicitly linked with representations of more abstract concepts and relationships in the problem domain to provide even deeper domain understanding.

C. Graphical representation

Examples of graphical representations of the problem domain identified by participants include flow charts, various types of diagrams (e.g. process flow diagrams, work flow diagrams, and system structure diagrams), and UML diagrams (e.g. class diagrams and sequence diagrams), as well as high level architecture or module diagrams. Generally participants did not distinguish between domain- or solution-oriented representations from this list.

The general view of the participants is that the structure inherent in the notation of most of the graphical representations lessened the likelihood of ambiguity compared to natural language representations. Most Interviewees stated that the use of graphical representations could be initiated by either the vendor or the client groups, However it tended to be the vendor group. Some participants further noted that the graphical representations tended to be used more within the vendor group for sharing understanding because they believed that the clients generally didn't have the skills to "use" these representations. Typically graphical representations were used *after* some degree of understanding has been elicited and were used as a conceptual "snapshot" of understanding (and not always kept current). For example, some participants reported using such graphical representations in their specifications documents. Furthermore, some participants' describe experiences of using graphical representations in a more spontaneous (and less persistent) way such as drawings on a white board during a discussion session between the vendor and the client groups.

From the data illustrated in table 4.6, it can be seen that graphical representations appear to be the least used by practitioners, particularly UML diagrams. This is explained by participants by the client group's lack of familiarity with the notation and meanings of such representations. They therefore perceive them as having limited value in sharing, refining and confirming domain understanding with the client group. If they are used, it is agreed that natural language (written or spoken) explanations and commentary are needed to supplement the graphical representation.

In the literature (for instance: Offen, 2002), graphical representations, particularly UML, are often recommended for modelling the application domain. These recommendations, however, need to be considered in the context of their use within *development* teams to model and understand application domains. Several empirical studies (e.g. Karlsson, 2007) point out the ineffectiveness of UML representation when used to represent knowledge to share with client groups.

Perhaps there is a need for a graphical notation that is familiar to clients and can be manipulated for "testing" by clients, but can automatically be mapped to more formal graphical constructs for use by designers and developers (and vice versa).

4.3.6. Perception of importance in developing shared understanding

One way of gauging practitioners' levels of motivation for process improvement in the area of domain understanding is to survey their perceptions regarding the importance and efficacy of activities and practices related to developing this understanding. One of the interview questions (question no.37, see Appendix A), therefore, explicitly asked the participants how they viewed the importance of developing shared domain understanding. They were asked to select a value of 1 to 5 on a Likert scale, with 1 representing “not at all important” and 5 ”very important”. Participants were unanimous in selecting the highest importance score (5), indicating a clear consensus on the importance of developing shared domain understanding.

It is interesting to compare this stated high degree of importance with the claimed level of effort on activities that develop shared understanding. This would provide some idea of whether this perceived importance converts into action. The estimated efforts and other significant efforts identified from each organisation are presented in table 4.7.

Org.	% of total project effort allocated for developing SU	Other identified significant efforts
1	***50%***	Domain expert employment
2	***20%***	Regular meetings to assess current understanding
3	***70-80%***	Regular meetings (workshops)
4	***50%***	Not clear
5	***40-50%***	Change of development methodology, which focus more on communication
6	***20%***	Professional RE training to improve RE practice and organizing RE Forum within the organisation
7	***20%***	Not clear
8	***20%***	Regular weekly progress meeting with the client group
9	***30%***	Regular communication and professional communication training for the staff
10	***30%***	Business people employment (as business analyst)

Table 4.7 – Organisations’ efforts in developing shared understanding

It can be seen that for more than a half of the participating organisations, 20-30% of total project effort is claimed to be spent in developing shared understanding. Four of the organisations do claim that half or more of the project effort (and up to 80%) is related to sharing problem domain understanding. Those with stated low efforts discuss the need to maintain the balance of shared understanding development activity with other activities, such as: development, testing and implementation.

There are several factors that may influence the stated effort of different organisations which need further investigation. For example an organisation's business domain may be a factor. An instance of this may be organisation 3, which claims a very high level of effort. This organisation's business is largely business process management and most of their effort goes into understanding their clients' business processes, which they classify as contributing to sharing domain understanding. In addition, the development activities (creating prototype model of business process management plan) are also closely identified with the development of shared understanding

An organisation's software offering, as either a software service or a product-driven company, may also be a factor in perceived effort. Product driven organisations are found to have more sources of knowledge to evaluate (section 4.3.3) and generally need to consider the requirements of a wider scope of clients or users, and so may be expected to have a perception of high effort in sharing understanding.

Analysis of the data presented in table 4.7 also indicates differences in dimensions that become the focus in terms of supporting the development of shared understanding. Information on the significant efforts are selected based on several criteria such as: the frequency of occurrence during the interview (e.g. organisation 2, 3, 8, and 9's regular communication with the client group), special emphasis by the participants (e.g. organisation 1's domain expert employment, organisation 5's change of methodology, organisation 10's business people employment), or it appears to be an exception compared to the other organisations (e.g. organisation 6's RE training, organisation 9's investment on communication training).

From table 4.7, it can be seen that there is quite a balance of focus between the vendor group stakeholder's quality (e.g. domain expert employment (organisation 1, 10) and trainings for improving vendor stakeholders' skills (organisation 6 and 9) and regularity of the communication (organisation 2, 3, 8, and 9).

4.4. Discussion

This section presents a discussion of several themes that emerge from the data analysis. This section also includes a discussion of the implications for either practice or research.

4.4.1. Quality of the client representatives

Analysis of the problems or barriers faced by the case organisations in developing shared understanding with their client suggests that there are certain qualities from the client representative that seems to be important for supporting the development of shared understanding between the two groups. In general, qualities that are perceived by the case organisations (from the vendor group) can be categorized into several dimensions, namely: (1) client representative's level of understanding on the domain knowledge, which has a strong relationship with the quality and stability of the knowledge or feedbacks communicated to the vendor group; (2) client representative's level of authority which, for instance, is needed for decisions that need to be made regarding the shared knowledge ;and (3) client representative's willingness to support the development of shared understanding by providing positive contribution, including providing sufficient commitment to the project, being cooperative in providing feedback, and being reasonable.

Suppose that all of these qualities are available within the client representative, it would be very likely to improve the performance of shared understanding development. However, result on the barriers in shared understanding development suggests that it is very difficult to realize all of these qualities in a client representative, as evidenced in many problems related to the client representatives' quality (e.g. insufficient domain knowledge, uncooperative client, overly demanding client, and so on) (see section 4.3.1). This may suggest that there are certain factors that influence the client group's way for selecting their representative, which may perhaps relate to their technique in selecting their representative. Since it is beyond the scope of this research to investigate the way client groups choose their representative, this might provide another point to be investigated further.

In summary, all of these dimensions are illustrated in figure 4.2. The identification of these dimensions could provide researchers with a framework for examining the client representative's quality in the context of developing shared problem domain understanding.

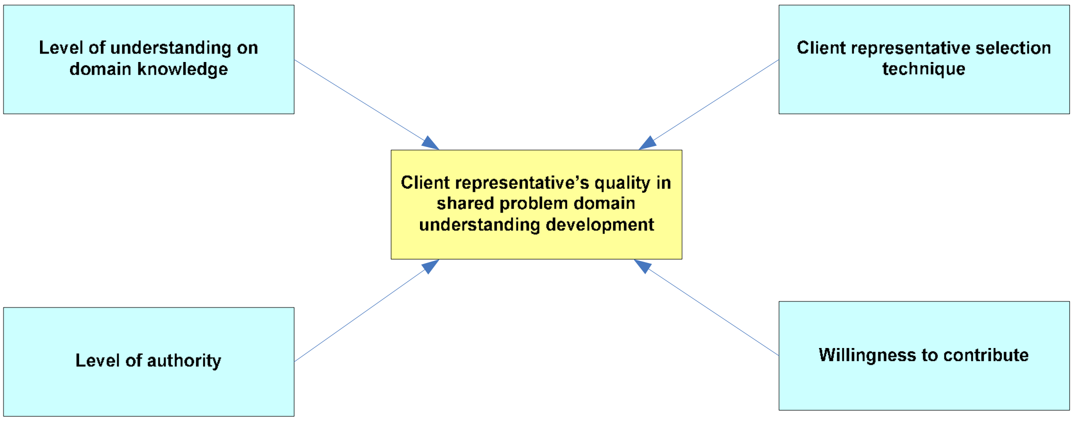
Level of understanding on domain knowledge
Client representative selection technique
Client representative's quality in shared problem domain understanding development
Level of authority
Willingness to contribute

Figure 4.2 – Client representative's quality framework

4.4.2. Level of understanding

It is noted from the interviews with the participants that different level of understanding on the problem domain exists among the vendor stakeholders. This is evidenced, for instance, by some organisations limiting (or even forbidding) the developers from taking part in knowledge sharing activities at different phases of development. More evidence is provided by one of the participants who specifically suggested that it is best to involve developer only when the communication with the client representative talks about work flow, as developers often have difficulty in understanding business processes. This suggests that certain level of understanding is sufficient for certain roles within the vendor group in order to progress the project.

Commonly, this level of understanding is linked with dimensions such as the depth and breadth of understanding on the problem domain. However, this would lead to further questions regarding the vendor stakeholder's understanding on when to recall their knowledge of the problem domain and how to apply them during the development phases. Other interesting aspect about their level of understanding perhaps can be related to their ability to abstract their current knowledge to understand the "big picture" related to their current understanding. All of these dimensions are illustrated in figure 4.3 below. Exploration to these dimensions would provide another area for future research which might improve our understanding on how to measure the level of shared understanding as well as improving our understanding about the importance of developing shared understanding particularly for the vendor group.

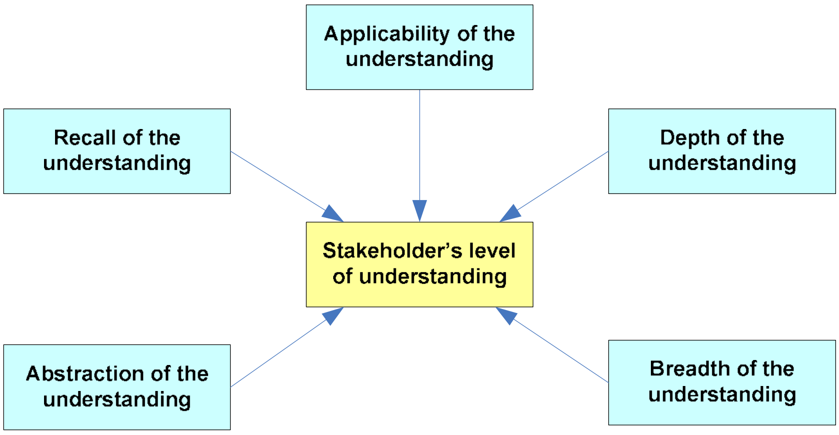
Applicability of the understanding
Recall of the understanding
Depth of the understanding
Stakeholder's level of understanding
Abstraction of the understanding
Breadth of the understanding

Figure 4.3 – Stakeholder's level of understanding dimensions

4.4.3. Representation evaluation metrics

Examination on the interview transcripts also identified several aspects regarding the representation of understanding that are considered significant by practitioners. Some of these aspects were not explicitly mentioned by the participants, but subject to interpretation based on the participants' comments during the interviews.

The first aspect on the representation concerns the universal characteristic of the representation. In the context of this study, this universal characteristic is linked with the readability of a representation by different stakeholders with diverse characteristic (e.g. skill, first language used), which make the understanding and interpretation process not difficult. One common suggestion from the result is by employing something "visual" or "pictorial" which typically can be found in a prototype or screen shot.

The second aspect is the rich characteristic of a representation. Commonly this richness is linked with the representation's ability to cover the significant breadth and depth of knowledge. For example: natural language is considered very rich because it can represent most concepts and relationships in most domains. Perhaps, in some respects, this characteristic can be linked with the ability of a representation to trigger the elicitation of tacit knowledge, such as a user's working habits. However, this particular characteristic still needs to be confirmed through further investigation.

Analysis of data on the barriers also indicates that the ability of the representation to engage the stakeholders' participation can be a significant factor as well. This engagement typically relates to the attractiveness of a representation and also a representation's ability to encourage discussion between the vendor and the client groups. Lastly, another aspect that might be relevant is the effort required in adopting the representation. This can be related to cost, training, or any other effort that is necessary (Palshikar, 2001). However, interestingly, this aspect was not discussed by any participants during the interview.

The identification of these factors (which are summarized in figure 4.4 below) could provide a framework for evaluating the efficacy of a representation of knowledge, which is not limited to representation of problem domain only.

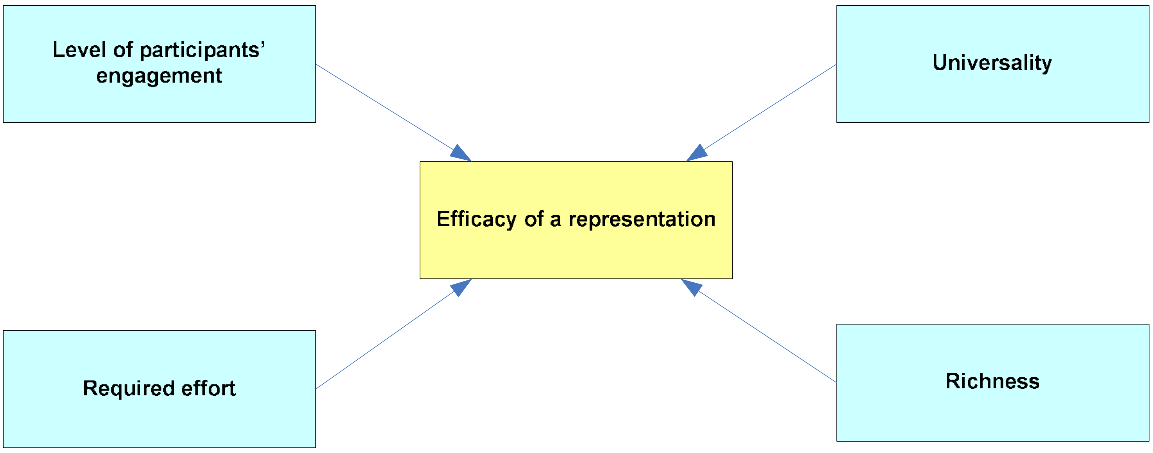
Level of participants' engagement
Universality
Efficacy of a representation
Required effort
Richness

Figure 4.4 – Efficacy of a representation

4.5. Conclusions

This chapter presents the main findings of this field study research. This chapter also includes analyses of the results and some discussions regarding the implications of this research. The research conducted can be considered wide in scope, as it focuses on various dimensions relating to development of shared understanding that have been identified from the literature. These various dimensions consist of:

- Barriers in developing shared understanding along with the ways to address the barriers,
- Stakeholders involved, in terms of role within the organisation,
- Processes in developing shared understanding,
- Tools and techniques along with their efficacy,
- Representations of understanding,
- The participants' views on the importance of developing shared understanding

All of these dimensions are explored in more details based on perceptions of the representative of the ten participating organisations. Moreover, comparisons of the findings to existing literature were also applied whenever possible. The chapter concludes with a discussions of some interesting aspects related to development of shared problem domain understanding with some possible frameworks suggested for future testing.

The next chapter addresses the conclusion of this thesis. It relates the findings explicitly to the research objectives and research questions, as well as reflecting on limitations of the study. Some implication for practice and possible future research in this area are discussed at the end of this concluding chapter.

Chapter 5

Conclusion

5.1. Introduction

This chapter provides a summary of this thesis along with some recommendations both for practice and future research in the area of shared problem domain understanding. Before describing the main conclusions of this study and relating them to the original research aims and questions, a brief review of the research aims, design and scope is provided in the next section. Section 5.3 presents an overview of the main findings of this research and relates them clearly to the research questions and framework of analysis. The implications of the findings of this study for practice are discussed in section 5.4. Section 5.5 provides a discussion of the limitations of this study and describes some conscious design decisions as well as reflections on lessons learnt. Finally, in the last section, some possible future research areas are explored.

5.2. Research Design

The main motivations for conducting this study of the phenomenon of sharing domain understanding in software development is based on an extensive literature review and the author's software development experiences and is summarised in the following observations:

- The clear importance of shared understanding development as a significant factor in improving the quality of software development, in particular collaborative activities such as found in requirements engineering.

- The difficult and inconsistent nature of sharing domain understanding in software development projects. This indicates the scope for improvement of such activities through better understanding of the process, the actors and the barriers, leading to the development of improved supporting tools and techniques that might mitigate some of the barriers.
- The lack of empirical research on sharing domain understanding in the current literature, particularly in the context of software requirements engineering.

These provide a strong rationale to explore the process of shared understanding development in order to obtain deeper understanding of this phenomenon. Since requirements engineering is commonly associated with communication activities between vendor and client to develop shared understanding of the problem domain (Nuseibeh & Easterbrook, 2000), this is selected as a context for this study.

Based on this motivation, the objectives of this research are described as:

- To identify and analyze the activities that are relevant to shared understanding development, including their frequency and who is involved in them.
- To identify tools and techniques used that potentially support the development of shared understanding, as well as gauge their efficacy.
- To identify the types of representation used to represent the knowledge of problem domain that needs to be shared.
- To investigate practitioners' perception of importance regarding the development of shared understanding, as well as their effort allocated to related activities.
- To identify barriers that would hinder practitioners from developing shared understanding, as well as their methods for overcoming these obstacles.
- To compare what is being reported in literature and current practice, in order to identify gaps or provide supporting empirical evidence.

In order to meet these research objectives, some research questions are formed to investigate the issues embodied in the research objectives. Research objectives and the research questions are summarized in table 5.1 along with the relationship between them.

Research objectives	Research questions
To investigate practitioners' perception of importance regarding the development of shared understanding, as well as their effort in achieving such a condition.	Do stakeholders perceive that developing shared understanding problematic (or difficult)? If so, what do they say the barriers are?
To identify barriers that would hinder practitioners from developing shared understanding, as well as their methods of overcoming these obstacles.	
To identify and analyze the activities that are relevant to shared understanding development, including their frequency and who is involved in them.	What do stakeholders do (processes, techniques, tools, and representations) in order to develop shared understanding of the problem domain with their client?
To identify tools and techniques used that potentially support the development of shared understanding, as well as their efficiency.	
To identify the type of representation used to represent the knowledge of problem domain that needs to be shared.	
To compare what is being reported in literature and current practice, in order to identify gap if one is present and possible explanations for it.	How does what the stakeholders say they do compare to what is reported in research literature?

Table 5.1 – Link between the research objectives and research questions

The selection of the 10 case organisations that participated in this study was limited to Auckland-based organisations whose main business is software development, both software service (bespoke) and product-driven. Although it is acknowledged that the clients' perspectives should provide further insights and in some sense triangulate the vendors' perspectives, this study is only considering the vendor group's perspective. This is partly due to uncertainty in the required time frame for this study (the data gathering and transcription phases took longer than expected). Also it was reasoned that it is more likely that the vendor group will put more effort in developing shared problem domain understanding because of the motivation of a "sale" and a satisfied client.

Due to the interest of this study in the perspectives of its participants, an interpretive approach to the data analysis is chosen and a multiple-case study is selected as the research method. Semi-structured interviews are chosen as the data-collection method and the interviews transcribed and qualitatively analysed using coding and categorizing techniques to identify emerging trends and patterns along several dimensions, identified from literature as significant.

The research design has suited this exploratory research and has provided some insights into the process of sharing domain understanding that suggest some directions for research into improvements to practice. These conclusions are the subject of the next section.

5.3. Conclusions

In this section, conclusions regarding the main findings of this research are drawn, structured according to the framed research questions.

5.3.1. Do stakeholders perceive that developing shared understanding is problematic? If so, what do they say the barriers are?

Based on the interview result, twelve barrier related themes are identified from the interview transcripts. All of these barriers are listed in table 4.2 (chapter 4). The identification of these barriers within the participants' experience in developing shared understanding suggests that majority of the case organisations perceive that developing shared understanding with their client is problematic and this problems are mainly related to people, the communication process, as well as the representation used to communicate understanding. However, the majority of case organisations seem to put more emphasis on peoples' characteristic that can affect the quality of the communication process (e.g. sufficient domain knowledge, uncooperative manner, etc.).

Further analysis on the barriers also indicates that quality of the client representation plays an important role in the process of developing shared understanding. This importance is particularly related to the level of their domain knowledge understanding, their authority in making decisions related to alternatives in the problem domain, as well as their willingness to contribute positively (see figure 4.1). With many emerging problems related to client representation quality, it may suggest that there are problems with the way client group selecting their representative, which seems to require further investigation.

5.3.2. What do stakeholders do (processes, techniques, tools, and representations) in order to develop shared understanding of the problem domain with their client?

A. Stakeholders involved

Figure 4.1 (chapter 4) presents the summary of possible roles involved in the process of shared understanding development between vendor and client group. It is apparent from this that stakeholders with the role of “analyst” can play an important part in sharing domain understanding by acting as the main “channel” of knowledge between vendor, client and regulator stakeholder groups. Findings of this study indicate that it is important for the “analyst” to have at least basic level of domain expertise in order to comprehend the (problem) domain being shared by the client group. This domain expertise is generally found in a domain expert employed by the vendor group to take the role of “analyst”. In other cases, it is obtained from experiences in doing projects or through personal research.

B. Process

One of the interesting findings from the analysis of the process described by participants is that regular and frequent communications between the client and vendor groups is often a key factor in developing shared understanding between the two groups. This regularity relates to domain knowledge elicitation, as well as verification and validation activities.

The frequent communications may be scheduled or ad-hoc interactions at varying level of formality and time effort. Participants acknowledge that shared understanding is a gradual process of evolution requiring regular and iterative interactions throughout software development. Regular communication is perceived to be effective for identifying issues that emerge during the project, as well as for identifying misunderstandings (e.g. related to jargon or ambiguity). This relationship with regular interactions and depth of shared understanding is further strengthened by those organisation which lack regular communications with their client as considering their current practice of shared understanding development as below average.

In some case organisations, the necessity of face-to-face communication is emphasised because of its richness and immediacy. Other modes of communication (e.g. e-mails, phone calls, and web-conferences) are also often mentioned in relation to the importance of this regularity in communication, which suggests that every chance for communication seems to be significant for developing shared understanding.

Two significant differences between software service organisations and product driven organisations, in relation to the process activities, can be identified:

- Product-driven organisations elicit knowledge from a wider range of stakeholders (e.g. various user groups) compared to software service organisations. This is often facilitated by conducting focus groups or employing tools that can be accessed widely and can gather a wide range of inputs, such as web-based prototypes, issue tracking systems, and on-line forums.
- In contrast to software service organisations, product-driven organisations often don't have an easily identifiable client to provide feedback on the current state of understanding, since their product is intended for a market. This role of "feedback provider" is usually fulfilled by a significant figure, such as an internal or external domain expert, or a trusted existing customer.

C. Tools and techniques

In terms of tools and techniques, conversational techniques (e.g. interview, conversation either formal or informal) and written documents are the most common techniques and tools used during the development of shared understanding

Another category of tool that is common is prototype-based tools, which can include parts of a product, prototype of a system, or screen shots). The most common benefits suggested by the participants are the "visual" and "tangible" characteristics of this type of representation. This can potentially reduce barriers such as unfamiliar language (pictures are generally more accessible) or the inability of stakeholders to assess at a higher conceptual level.

Participants also identify important factors may influence the selection of certain tools or techniques including, the client's business domain, previous experiences with tools, and the skill level of the client stakeholders. This suggests that it may be difficult to generalize the efficacy of tools or techniques in terms of shared understanding development due to the variety of circumstances faced by each organisation.

D. Representation

Findings on the representation of understanding indicate that natural language and product type representation seems to be popular among the case organisations. This means that there is a trend for selecting types of representation which are considered "universal" or in other words can be understood easily by majority of the involving stakeholders from both groups.

Natural language representation can be understood easily, generally due to its nature of being employed in daily lives by majority of people and it apparent that it is being used widely across the process of developing shared understanding (in the early phase when the problem domain is elicited until the verification and the validation phase). On the other hand, product type representation is comprehensible due to its "visual" characteristic which shows something "tangible", "concrete", or "real". This characteristic particularly perceived as having the ability to trigger discussion that can enhance the generation of idea (during elicitation phase) or improve the stakeholders' understanding on the knowledge being represented (during verification and validation phase).

It is also apparent that the employment of representation other than natural language (e.g. product type representation, graphical representation) seems to be complemented by the existence of natural language representation itself, either written or verbally. This suggest that natural language representation is the main key for representing understanding, as other type of representation (perhaps) can be regarded as a complementary in order to mitigate natural language's ambiguous nature.

5.3.3. How does what the stakeholders say they do compare to what is reported in research literature?

In general, most of the participants' experiences are found to align with theory and practice reported in literature (particularly from RE related literature (e.g. Cao & Ramesh, 2008; Coughlan et al., 2003; Emam & Madhavji, 1995; Saiedian & Dale, 2000). This similarity along each of the dimension of analysis can be seen in the detailed discussion presented in chapter 4. The contemporary view of shared understanding and requirements being emergent from iterative and collaborative activities prevails significantly over the more traditional waterfall and "sign-off" view (see for example Alcázar & Monzón, 2000; Damian, 2007; Desphande et al., 2005; Rickman, 2001). The importance of the characteristics of the client representative(s) is emphasised more in this study than is found in literature.

The categories of barriers related to sharing understanding identified in the study generally match those found in literature. It is clear that because of the diversity of each organisation's circumstances, one suggestion for handling a particular barrier may not be applicable in other case organisations, thus there is variability in the efficacy of a method.

In terms of tools, it is found that RE tools such as: Rational Requisite Pro and Telelogic Doors, which are recommended in literature for supporting interactive collaboration between stakeholders in software projects (e.g. Damian, 2007), are not identified at all in participants' experiences. It is likely that this kind of tool is not popular for SMEs due to the effort (e.g. cost, training) needed to implement such a tool.

In contrast to suggested practices reported in literature (see for example Zowghi & Coulin, 2003) participants in this field study readily identified prototype-related techniques as improving the development of shared problem domain understanding (section 4.3.5).

5.4. Implications for practice

As described and discussed in chapter 4, the quality of the stakeholders involved in sharing understanding is a significant factor in the efficacy and efficiency of the related activities and process. The frequent discussions of instances of poor representatives suggest that there may inadequate techniques and processes for selecting high quality representatives, particularly client representatives. Figure 4.2 in chapter 4 provides practitioners with some dimensions that are considered important by practitioners in developing shared understanding. However, if the opportunity for selecting the client stakeholder representative(s) is limited, the success in developing shared understanding depends largely on the vendor group representative's communications skills and domain expertise. Therefore, it is recommended that the vendor group develops their staff in the analyst role in terms of sufficient communication skills and domain expertise, by appropriate training if necessary

Another implication relates to the importance of having regular communication during the project in order to develop shared understanding. This enhances shared understanding by:

- Elevating the possibility of detecting misunderstandings or misinterpretations between the client and vendor groups.
- Opening opportunities to reduce the workload of developing shared understanding at a single session and giving the analysts the opportunity to assimilate new knowledge and refine understanding.
- Maintaining stakeholders' current state of understanding, since continuous regular meetings should shape a solid understanding of the problem and also "refresh" the memory of the stakeholders.

In order to realise this regularity, it is highly recommended that the vendor group obtains the client group's commitment to the process as early as possible during a project. This can be done by getting agreement on its importance at the early phase of a development project. Furthermore, increasing the frequency of informal communication such as phone calls, emails, and casual meetings can improve the frequency of interactions that could lead to refinement of domain understanding.

Another implication relates to the importance for having domain expertise within the vendor group. This can provide some advantages in terms of:

- Mitigating the effect of jargon usage during communication between vendor and client groups.
- Providing expert's input or validation when there is a difficulty in accessing the client group's stakeholders.

This can be realized by employing a domain expert. Domain expertise development can also be obtained through training, involvement in projects, or studying existing project documentation. However, compared to employing domain experts, these approaches are a trade-off between the time and cost.

The importance of developing partnership with client stakeholders, especially with someone in an influential position within the client organisation, also has implications for practice. Developing this partnership can be done by maintaining regular communications with the client stakeholders. Developing relationships with senior members of the client hierarchy could potentially be beneficial in developing shared understanding by:

- Opening the opportunity to seek replacement if there is a problem with the current representatives.
- Ensuring there is access to a stakeholder with decision making authority, which may be necessary if there are multiple points of view.
- Opening the opportunity to know the client representative's characteristics that might be useful in developing shared understanding, such his/her skills, domain expertise, and personality traits.

Having identified some practical implications of the findings of this investigation, the next section discusses some limitations in the design and implementation of the study.

5.5. Limitations

Reflecting on the design and execution of this study provides some lessons learnt, some shortcomings and some possible extensions to the research.

The first limitation of this study concerns the fact that each interview provides a viewpoint limited to the given representative's perspective. It is possible that other roles within the organisation would have different points of view regarding shared understanding development that might enhance the richness and validity of the data. Details on the participants (table 3.5) indicate that this research has not taken the views of certain roles related to software development, particularly developers, who tend to obtain the domain knowledge from senior to high-level management.

The second limitation of this study concerns recommendations from experts regarding the importance of "data triangulation" in case study research (e.g. Tellis, 1997; Yin, 2003), to improve the validity of the research. Data triangulation usually consists of comparisons between data obtained from different sources such as interviews, document analyses and observations. This research uses only semi-structured interviews as a data collection method, with little effort at triangulation from multiple data sources. However, the result of this research has provided information on what kind of documents need to be analyzed (e.g. client's white paper, project proposal, requirements template, e-mail, etc.) and also parts of activities that need to be observed (e.g. product demonstration session, initial project meeting, workshops, training session, etc.). This information is useful when planning similar future research on shared understanding development.

Another limitation became apparent during data analysis. The use of Likert scales in the questions on the perceptions of the efficacy of tools and techniques was largely ineffective. These scales were often just a distraction to the unanticipated excessive amounts of information given by the participants related to these questions. This often resulted in the researcher missing some of the tools or techniques mentioned by the interviewees, thus results in incomplete acquisition of the data.

Lastly, all of the data collected during the interviews are the participants' perceptions and it is possible for the data to be interpreted differently to their original meaning by the researcher, thus threatening the validity of the data. Yin (2003) recommends that relevant participants should review the data before they are recorded in the final report. However, due to unexpected time pressures this activity was not undertaken to any degree.

As is common in research, this study has raised more questions than it has answered and some fruitful areas of further research related to this study are discussed in the next section.

5.6. Future research

As mentioned in chapter 1, a future study focusing on the client group's perceptions of developing domain understanding (ideally with these same vendors) is considered a useful extension to this study. This investigation may be useful for identifying differences between the client and vendor groups, thus leading to other relevant insights that maybe useful for improving the current practice of shared understanding development. Moreover, a study on client group's perception may also open the opportunity to verify the vendors' perceptions of client activities such as the method employed by the client group in validating vendor group understanding; and factors that influence the client group in selecting their representative.

Reflection on the demographics of this study indicates that this study lacks input from the developer group, which is clearly an important role in the development of software. It is very likely that developers' perspectives would provide further insights to the practice of shared problem domain understanding. Thus, further investigation that incorporate wider participation (including the developers) from each case organisations can be considered as one possible future research area.

Result from the literature review identifies several aspects that are relevant to the development of shared problem domain understanding practice, namely: stakeholders, processes, activities, tools /techniques and representations (see figure 2.5). The relevance of these aspects have been confirmed in this study, thus provide a framework for similar future research in this area. However, the level of influence brought by each of these aspects to the development of shared problem domain understanding practice is still not clear. Investigation on this matter would provide practitioners with insights in determining their project management plan, particularly in the practice of shared understanding development.

In terms of tools, techniques and representation, figure 4.4 provides a framework for determining the efficacy of a representation used for communicating knowledge. Besides providing a framework for future research on the area of knowledge representation, dimensions on the framework also offer some criteria that can be useful for developing new tools or techniques that can support knowledge representation as well as shared understanding development activities.

Another opportunity for future research also comes from the popularity of prototype-based techniques and representations in the shared problem domain understanding development process. Since, there is little information regarding the relationship between problem domain understanding and the use of prototype-based techniques and representation, further research to clarify and understand this relationship might be useful.

Lastly, similar empirical studies of practice in a broad range of domains are desirable. As one participant expressed it, they are “keen to be exposed to other experiences” to improve their own practice in this area of developing shared problem domain understanding. This may suggest that research in this area is considered very important by practitioners and needs to be socialized widely.

References

Adler, P. A., & Adler, P. (1998). Observational techniques. In N. K. Denzin & Y. S. Lincoln (Eds.), *Collecting and interpreting qualitative materials*. Thousand Oaks: Sage Publications.

Al-Rawas, A., & Easterbrook, S. (1996). Communication problems in requirements engineering: A field study. *Proceedings of 1st Westminster Conference on Professional Awareness in Software Engineering, Royal Society, London, 1996.*

Alcázar, E. G., & Monzón, A. (2000). A process framework for requirements analysis specification. *Proceedings of 4th International Conference on Requirements Engineering*, 2000, 27-35.

Alexander, I., & Robertson, S. (2004). Understanding project sociology by modeling stakeholders. *IEEE Software*, *21*(1), 23-27.

Aranda, J., Easterbrook, S., & Wilson, G. (2007a). Requirements in the wild: How small companies do it. *Proceedings of the 15th IEEE International Requirements Engineering Conference (RE'07)*, Delhi, India, 2007.

Aranda, J., Khuwaja, R., Easterbrook, S. (2007b). Discovering the shared understanding dynamics of large software teams. *17th International Conference of the IBM Centers for Advanced Studies (CASCON'07)*, Toronto, Canada.

Arthur, J. D., & Gröner, M. K. (2005). An operational model for structuring the requirements generation process. *Requirements Engineering, 10*, 45-62.

Aurum, A., & Wohlin, C. (2005). Requirements engineering: Setting the context. In A. Aurum & C. Wohlin (Eds.), *Engineering and Managing Software Requirements*. Berlin: Springer Berlin Heidelberg.

Berry, D. M., & Kamsties E. (2003). Ambiguity in requirements specification. In J. C. Samapaio do Prado Leite & J. Horacio (Eds.), *Perspectives on software requirements*. Springer.

Berzins, L. V., Shing, M., Riehle, R., & Nogueira, J. (2000). Evolutionary computer aided prototyping system (CAPS). *Proceedings of 34th International Conference on Technology of Object-Oriented Languages and Systems*, 2000, 363-372.

Bowden, J. A. (2000). The nature of phenomenographic research. In J. A. Bowden & E. Walsh (Eds.). Phenomenography. Melbourne: RMIT University Press.

Boyatzis, R. E. (1998). *Transforming qualitative information*. California: Sage Publications.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology, 3*, 77-101.

Cao, L., & Ramesh, B. (2008). Agile requirements engineering practices: An empirical study. *IEEE Software*, *25*(1), 60-67.

Cannon-Bowers, J. A., & Salas, E. (2001). Reflection on shared cognition. *Journal of Organizational Behavior, 22*, 195-202.

Carley, K. M., & Lawler, R. W. (1996). *Case study & computing: Advanced qualitative methods in the study of human behavior.* Intellect Books: Fishponds, Bristol, United Kingdom.

Cederling, U., Ekinge, R., Lennartsson, B., Taxén, L., & Wedlund, T. (2000). A project management model based on shared understanding. *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, 2000, 1-10.

Cohen, S. G., & Gibson, C. B. (2003). In the beginning: Introduction and framework. In C. B. Gibson & S. G. Cohen (Eds.), *Virtual teams that work: Creating conditions for virtual team effectiveness*. San Fransisco: John Wiley & Sons.

Collis, J., & Hussey, R. (2003). *Business research: A practical guide for undergraduate and postgraduate students*. New York:Palgrave MacMillan.

Coughlan, J., Lycett, M., & Macredie, R. D. (2003). Communication issues in requirements elicitation: A content analysis of stakeholder experiences. *Information and Software Technology, 45*(8), 1, 525-537.

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

Curtis, B., Krasner, H., & Iscoe, N. (1988). A field study of the software design process for large systems. *Communications of the ACM*, *31*(11), 1268-1286.

Damian, D. (2007). Stakeholders in global requirements engineering: Lessons learned from practice. *IEEE Software*, *24*(2), 21-27.

Davies, I., Green, P., Rosemann, M., Indulska, M., & Gallo, S. (2006). How do practitioners use conceptual modeling in practice?. *Data and Knowledge Engineering, 58*, 358-380.

Decker, B., Ras, E., Rech, J., Jaubert, P., & Rieth, M. (2007). Wiki-based stakeholder participation in requirements engineering. *IEEE Software*, *24*(2), 28-35.

Deshpande, N., De Vries, B.,& Van Leeuwen, J. P. (2005). Building and supporting shared understanding in collaborative problem-solving. *Proceedings of Ninth International Conference on Information Visualisation*, 2005, 737–742.

Dubé, L., & Paré, G. (2003). Rigor in information system positivist case research: Current practices, trends, and recommendations. MIS *Quarterly, 27*(4), 597-635.

Emam, K. E., & Madhavji, N. H. (1995). A field study of requirements engineering practices in information systems development. *Proceedings of the Second IEEE International Symposium on Requirements Engineering*, 1995, 68-80.

Flynn, D. J., & Jazi, M. D. (1998). Constructing user requirements: A social process for a social context. *Information Systems Journal, 8(1)*, 53-83.

Fox, N. (2006). Using interviews in a research project. Retrieved July 07, 2008 from http://www.trentrdsu.org.uk/uploads/File/Using%20Interviews%202006.pdf

Genzuk, M. (2003). A synthesis of ethnographic research. Retrieved April 10, 2008 from http://www-rcf.usc.edu/~genzuk/Ethnographic_Research.pdf.

Gervasi, M. & Zowghi, D. (2005). Reasoning about inconsistencies in natural language requirements. *ACM Transactions on Software Engineering and Methodology*, 14(3), 277–330.

Gervasi, M., & Sawyer, P. (2008). Special section on natural language in software engineering. *IET Software, 2*(1), 1-2.

Glinz, M., & Wieringa, R. J. (2007). Stakeholders in requirements engineering. *IEEE Software*, *24*(2), 18-20.

Gordijn, J., Yu, E., & Van der Raadt, B. (2006). E-service design using i* and e^3 value modeling. *IEEE Software*, *23*(3), 26-33.

Hall, J. G., Jackson, M., Laney, R. C., Nuseibeh, B., & Rapanotti, L. (2002a). Relating software requirements and architectures using problem frames. *Proceedings of IEEE Joint International Conference on Requirements Engineering*, 2002, 137-144.

Hall, T., Beecham, S., & Rainer, A. (2002b) Requirements problems in twelve companies: An empirical analysis. *IEE Proceedings Software, 149*(5): 153-160.

Hammond, J., Rawlings, R., & Hall, A. (2001). Will it work?. Retrieved August 16, 2007 from http://www.praxis-his.com/pdfs/Will_it_work.pdf

Herbert, K. (2005). *Research methodologies in supply chain management.* Springer: New York, NY, USA.

Hilton, A. (2003). Engineering software systems for customer acceptance. Retrieved August 16, 2007 from http://www.praxis-his.com/pdfs/customer_acceptance.pdf

Hinds, P. J., & Weisband, S. P. (2003). Knowledge sharing and shared understanding in virtual teams. In C. B. Gibson & S. G. Cohen (Eds.), *Virtual teams that work: Creating conditions for virtual team effectiveness*. San Fransisco: John Wiley & Sons.

Hofmann, H. F., & Lehner, F. (2001). Requirements engineering as a success factor in software projects. *IEEE Software*, *18* (4), 58-66.

Hsieh, Y. (2006). Culture and shared understanding in distributed requirements engineering. *International Conference on Global Software Engineering*, 2006, 101-108.

Jarke, M. (1998). Requirement Tracing. *Communications of the ACM*, *41*(12), 32-36.

Karlsson, L., Dahlstedt, A. G., Regnell, B., Dag, J. N., & Persson, A. (2007). Requirements engineering challenges in market-driven software development – An interview study with practitioners. *Information and Software Technology, 49*, 588-604.

Klein, H. K., & Myers, M. D. (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly, 23*(1), 67-93.

Kudikyala, U. K., & Vaughn, R. B. (2005). Software requirement understanding using pathfinder networks: Discovering and evaluating mental models. *Journal of Systems and Software, 74(1)*, 101-108.

Lawrence, B. (1998). Designers must do the modelling. *IEEE Software*, *15*(2), 31-33.

Lu, C., Chu, W. C., Chang, C., & Wang, C. (2007). A model-based object-oriented approach to requirements engineering (MORE). *31st Annual International Computer Software and Applications Conference*, 2007, 153-156.

Lubars, M., Potts, C., & Richter, C. (1993). A review of the state of the practice in requirements modelling. *Proceedings of IEEE International Symposium on Requirements Engineering*, 1993, 2-14.

Maguire, S, Koh, S. C. L., & Huang, C. (2007). Identifying the range of customer listening tools: a logical pre-cursor to CRM? *Industrial Management & Data Systems, 107*(4), 567-586.

Maguire, S., & Ojiako, U. (2008). Market-led systemd development: When customers become users. *Industiral Management & Data Systems, 108*(2), 173-190.

Marton, F. (1986). Phenomenography: A research approach to investigating different understanding of reality. In R. R. Sherman & R. B. Webb (Eds.). *Qualitative research in education: Focus and methods*. London and New York: Routledge.

Mich, L., Franch, M.,& Inverardi, N. P. Market research for requirements analysis using linguistic tools. *Requirements Enginnering, 9*(1), 40–56.

Monarchi, D. E., & Puhr, G. I. (1992). A research typology for object-oriented analysis and design. Communications of the ACM, 35(9), 35-47.

Myers, M. D. (1994). A disaster for everyone to see: An interpretive analysis of a failed is project. *Accounting, Management and Information Technologies, 4*(4). 185-201.

Myers, M. D. (1997). Qualitative research in information systems. *MIS Quarterly, 21*(2), 241-242.

Neill, C. J., & Laplante, P. A. (2003). Requirements engineering: The state of the practice. *IEEE Software*, *20*(6), 40-45.

Nguyen, P. T., Babar, M. A., & Verner, J. M. (2006). Critical factors in establishing and maintaining trust in software outsourcing relationships. *Proceedings of International Conference on Software Engineering 2006*, pp. 624-627

Nikula, U., Sajaniemi, J., & Kälviäinen, H. (2000). A state-of-the-practice survey on requirements engineering in small-and medium-sized enterprises. Retrieved April 10, 2008 from http://www2.lut.fi/~unikula/Publications/TBRCRR1.pdf

Nuseibeh, B., & Easterbrook, S. (2000). Requirements engineering: A roadmap. *Proceedings of the Conference on the Future of Software Engineering*, Limerick, Ireland, 2000, 35-46.

Offen, R. (2002). Domain understanding is the key to successful system development. *Requirements Engineering*, *7*(3), 172-175.

Ojasalo, J., Nätti, S., & Olkkonen, R. (2008). Brand building in software SMEs. *Journal of Product & Brand Management, 17*(2), 92-107.

Orlikowski, W. J. (1991). Integrated information environment or matrix of control? The contradictory implications of information technology. *Accounting, Management and Information Technologies, 1*(1), 9-42.

Ottenheijm, S., Van Genuchten, M., & Geurts, J. (1998). What's the problem?. *Proceedings of the 31st Hawaii International Conference on System Sciences, 1*, 6-9.

Palshikar, G. K. (2001). Applying formal specifications to real-world software development. *IEEE Software*, *18*(6), 89-97.

Przybilski, M., & Tuunanen, T. (2007). From rich user requirements to system requirements. *Proceedings of 11th Pacific-Asia Conference on Information Systems*, Auckland, New Zealand, 2007, 561-573.

Rickman, D.M. (2001). A new process for requirements understanding. *The 20th Conference of Digital Avionics Systems*, 2001, *1*, 14-18.

Rooij, J. D., Verburg, R., Andriessen, E., & Hartog, D. D. (2007). Barriers for shared understanding in virtual teams: A leader perspective. *The Electronic Journal for Virtual Organizations and Network, 9*.

Rosencrance, L. (2007). Survey: Poor communication causes most IT project failures. Retrieved May 17, 2007 from http://www.computerworld.com/action/article.do?command=viewArticleBasic&taxonomyName=software&articleId=9012758&taxonomyId=18&intsrc=kc_top

Sadraei, E., Aurum, A., Beydoun, G., & Paech, B. (2007). A field study of the requirements engineering practice in Australian software industry. *Requirements Engineering*.

Saiedian, H., & Dale, R. (2000). Requirements engineering: Making the connection between the software developer and customer. *Information and Software Technology*, *42*(6), 419-428.

Segal, J. (2005). When software engineers met research scientists: A case study. *Journal Empirical Software Engineering, 10*(4), 517-536.

Seidman, I. (2006). Interviewing as qualitative research: A Guide for researchers in education and the social sciences (Third ed.). New York: Teachers College Press.

Sharp, H., Finkelstein, A., & Galal, G. (1999). Stakeholder identification in the requirements engineering process. *Proceedings of Tenth International Workshop on Database and Expert Systems Applications*, 1999, 387-391.

Shin, J. E., & Sutcliffe, A. G. (2005). Scenario advisor tool for requirements engineering. *Requirements Engineering, 10*(2), 132-145.

Tellis, W. (1997). Introduction to case study. *The Qualitative Report, 3*(2).

Tellis. W. (1997b). Application of a case study methodology. *The Qualitative Report*, *3*(3).

Thia, C. W., Chai, K., Bauly, J., & Xin, Y. (2005). An exploratory study of the use of quality tools and techniques in product development. *The TQM Magazine, 17*(5), 406-424.

Tsai, H., & Gururajan, R. (2007). Motivations and challenges for M-business transformation: A multiple-case study. *Journal of Theoretical and Applied Electronic Commerce Research, 2*(2), 19-33.

Tuunanen, T. (2003). A new perspective on requirements elicitation methods. *Journal of Information Technology Theory and Application, 5*(3), 45-72.

Voss, C., Tsikriktsis, N., & Frohlich, M. (2002). Case research in operations management. *International Journal of Operations & Production Management, 22*(2), 195-219.

Walsham, G., & Waema, T. (1994). Information systems strategy and implementation: a case study of a building society. *ACM Transactions on Information Systems, 12*(2), 150-173.

Yin, R. K. (2003). *Case study research: Design and methods* (Third ed. Vol. 5). Thousand Oaks: Sage Publications.

Zowghi, D., & Coulin, C. (2005). Requirements elicitation: A survey of techniques, approaches, and tools. In A. Aurum & C. Wohlin (Eds.), *Engineering and Managing Software Requirements*. Berlin: Springer Berlin Heidelberg.

Zowghi, D., Damian, D., & Offen, R. (2001). Field studies of requirements engineering in a multi-site software development organization: Research in progress. Retrieved May 18, 2007 from http://www.cs.uvic.ca/~danielad/AWRE/Zowghi_AWRE.pdf.

Appendix A – Interview questionnaire and protocol

A.1. Protocol

- **Researcher self introduction**. Name, school and faculty represented, purpose of the research.
- **Overview of interview procedure**: In this meeting, first we would explain about our research and then we'll continue with asking you some questions based on the questions that I sent you before, which will take about one to one a half hour.
- **Overview of the research project**: As for the research, the area that we interested in is in Requirements Engineering. In particular, as we showed in the model, one of the critical things in order to develop requirements that useful is that these two different groups (business/client group and developer group) somehow need to agree on a shared understanding of the domain where the project lies. It is important, as we can see from other projects that have been documented, that it is quite possible that developers might develop requirements which are different from what the client's thinking because they haven't developed their shared understanding together. Moreover, it's also often that the clients might change their requirements as they get better understanding. So we want to understand the process of developers in understanding the client's problem or business domain. We are particularly interested in what processes that the developers used to interact with their client in sharing their understanding and have agreement on it, and whether they used any specific technique or tool in the practice. So it has nothing to do with what the theories/academic said what should happen about them. We are interested in what actually happen. We also interested in identifying the barriers which make the process doesn't work, and how the developer and client manage the evolving understanding and requirements through the development project.
- **Reminder of some information:** Confidentiality issue, data stored procedure, etc.

- **Ask the participants whether they have further questions.**
- **Continue to the interview questions.**

A.2. Research Model

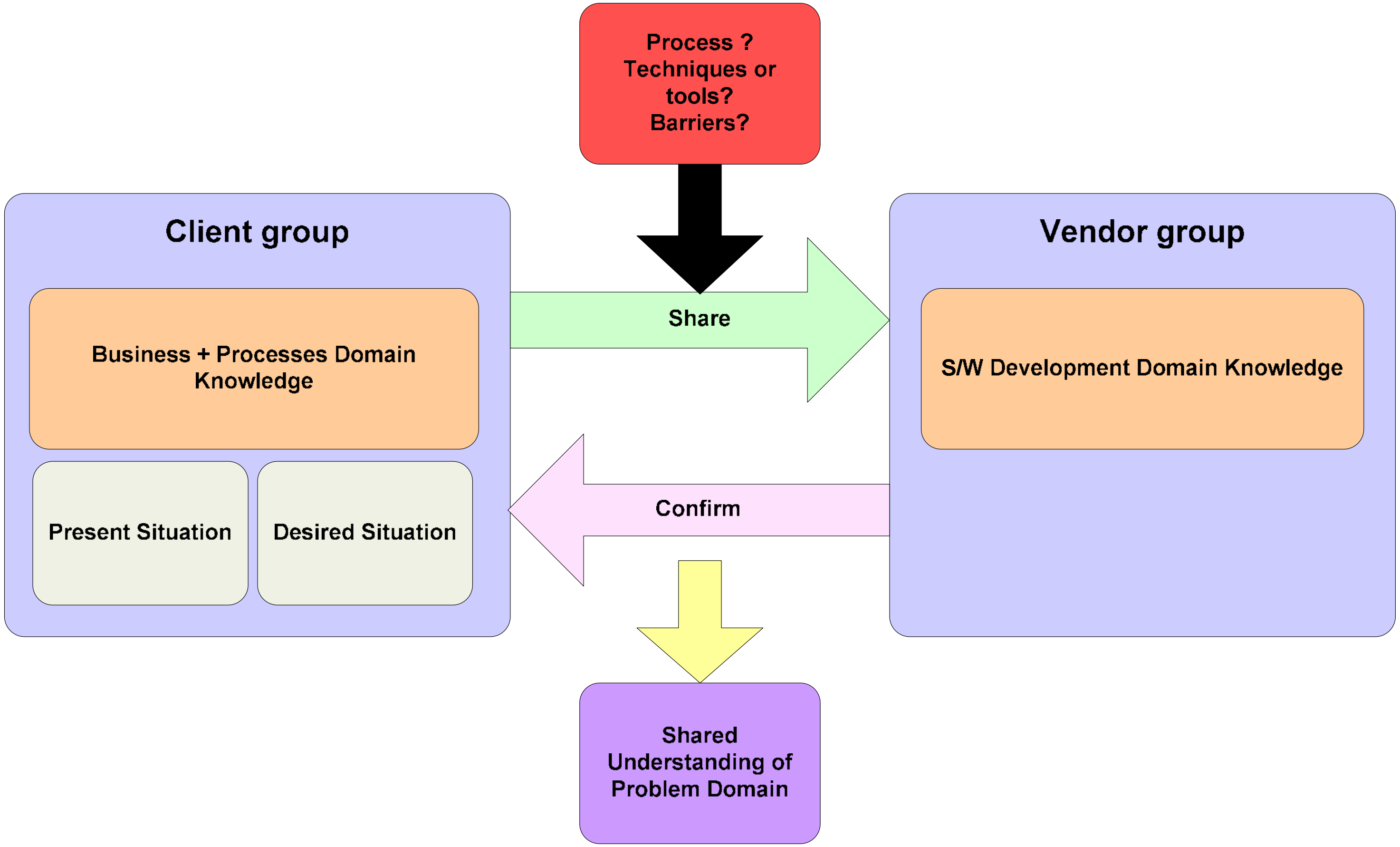

Figure A.1 - Shared understanding development model

A.3. Interview Questionnaire

1. Demographic questions

Initially, we just want to group our result based on some demographic information about the companies. So I would like to ask some demographic questions first. Your business domain is _________ right? Is there any other specific business domain that your organization into? ____

1. How many equivalent full time people does your organization employ?

   □ Less than 10 full time staff or equivalent

   □ 10 to 20 full time staff

   □ 21 to 49 full time staff

   □ 50 to 99 full time staff

   □ 100 to 300 full time staff

   □ More than 300 full time staff

   □ Do not know

2. What year was your organization established? _______

3. Approximately how many software projects is your organization involved in per year? _______

4. What is your title within the organization? _______

5. Approximately how many years of experience do *you* have in software development?

□ Less than 2 years

□ Between 2 and 5 years

□ Between 6 and 10 years

□ More than 10 years

6. How would you describe your role in your organization’s relationship with your clients? (e.g. consultant, etc.)

   ____

   ____

7. How would you describe your role in helping the developers and clients share understanding about the business problem or opportunity that the software will address?

   ____

   ____

2. Process related questions

2.1. General process

8. Does your organization follow a specific software development methodology (such as waterfall, Agile method, XP, etc., or any custom method)? If so, please describe it.

   ____

   ____

(***Note: Question no.9 will use Template no.1)***

9. For the last software development project, can you describe how your organization went about understanding the problem or opportunity that your client wanted to address? (By "how", I mean what activities, when, and who were involved).

Template no.1 (Question no.9)

Activities taken for understanding the problem domain	**Tools/ Techniques**	**When?**	**Who were involved?**

10. For this project, what things did you do that you think helped to share this problem domain understanding with your client?

____

____

(Note: Question no.11 will use Template no.2)

During some stages, developers might check whether they have the right understanding about what the client wants through something like prototyping or something else. So….

11. During this project, how did you confirm the development team's domain understanding with the client at any stage? (By "how", I mean what activities, when, and who were involved).

Template no.2 (Question no.11)

Activities taken for confirming the development team's understanding	Tools/ Techniques	When?	Who were involved?

(Note: Questions no.12-14 will use Template no.3)

12. For this last project, describe any specific techniques you used to help gain and share an understanding of the business problem domain with the client group?
13. Did you use any specific software tools (such as: database, spreadsheet, requirements engineering tools (e.g. DOORS, etc.), etc.) to assist in the process of getting and sharing an understanding of the problem domain with the client group?
14. Are the tools and techniques you described for this last project the usual tools/techniques you used for other software development projects? If not, can you describe the others?

(If the responses are not sufficient, continue with asking how the tools/techniques used?)

Template no.3 (Questions no.12-14)

Tools/Techniques	Type (Tools/Techniques)	How the tools/techniques used?

15. From the tools/techniques you have mentioned, which one is the best from your point of view? Why?

____

____

16. How many staff were involved in the last project we have been discussing? _______

17. How long did this project take to complete? _______

18. Was there anything unusual about this last project compared to other projects you have been involved with?

(For example: special customers that make you do more than usual, working with the client project manager, etc.)

____

____

2.2. Problems faced during the process

We have identified some of the processes, techniques, and tools those you use. Now, we would like to talk about some of the problems that you may encounter during these kin of processes. So….

(*Note: Questions no.19-22 will use Template no.4)*

19. For this last project, or any other projects, can you describe any barriers (or any other things that a bit hindered you in sharing the knowledge) that you find in sharing the knowledge about the business problem between you and your clients?
20. How did you handle those barriers?
21. What support from particular tool or technique did you use?
22. How effective do you think about the tool/technique you mentioned before?

Template no.4 (Questions no.19-22)

Barriers in sharing the knowledge	How do you handle them?	Tool/Technique used	Effectiveness (‘not effective at all’ to ‘very effective’) [circle one]	Why?
			1 2 3 4 5	
			1 2 3 4 5	
			1 2 3 4 5	
			1 2 3 4 5	

23. Would you say understanding the terminology or language related to your client's business area was a problem?

____

____

23a. If **Yes**, How do you deal with this?

____

____

23b. If **No**, How do you avoid it becoming a problem?

____

____

24. How well would you say about your development team and your client's team in communicating a problem? (circle one)

Not well at all **1 2 3 4 5** ***Very Well***

What sort of evidence can you give?

____

____

25. For the last project, to what degree do you think that your client's understanding of the business problem changed over the course of the project? (circle one)

Not changing at all **1 2 3 4 5** ***Changed dramatically***

25a. Do you think this is common in other projects?

____

____

25b. Was it a problem?

____

____

25c. How was the managing change process addressed?

____

____

(*Note: Questions no.26-28 will use Template no.5)*

26. Can you describe how you handled any disagreement in understanding of the business with your client?

____

____

27. Did you use support from particular technique or tool (e.g. database, software tool, etc.)? If so, please describe it and how it was used.

28. How effective are the tools/techniques?

Template no.5 (Questions no.26-28)

Tool/Technique used in handling conflicting ideas in developing a shared understanding	How was the tool/technique used?	Effectiveness ('not effective at all' to 'very effective') [circle one]	Why?
		1 2 3 4 5	
		1 2 3 4 5	
		1 2 3 4 5	
		1 2 3 4 5	

2.3. Recording understanding

29. For the last project, or any others, please describe how you documented or represented your understanding of the business problem or opportunity.

    ____

    ____

30. How did your client represent or document their understanding of the business problem?

    ____

    ____

31. How useful do you think this documentation of the business problem was useful for sharing understanding between your client and developers?

    Not useful at all **1 2 3 4 5** ***Very useful***

    Please explain.

    ____

    ____

32. How were business or user requirements for the software represented? (e.g. use cases, scenario, specification document, etc.)

____

____

2.4. Shared understanding

33. Can you describe how do you confirm that you have the same understanding as your client?

____

____

34. Can you describe how you know that the shared understanding you get is the same as the clients' understanding?

____

____

35. Do you have any criteria to measure the **'level of understanding'** and **'how well it is shared'**?

____

____

36. How would you rate the effectiveness of your organisation's current practice in developing shared problem understanding with the clients and the development team?

    Not effective at all*   1   2   3   4   5   *Very effective

____

____

37. How important do you think it is to develop and confirm a shared understanding of the problem domain during software development?

Not important at all **1 2 3 4 5** ***Very important***

37a. How important do you think the client thinks to develop a shared understanding of the problem domain during software development?

Not important at all **1 2 3 4 5** ***Very important***

37b. What degree of project effort would you say typically goes into developing a shared understanding?

☐ 10% or less from the total project effort

☐ 20% from the total project effort

☐ 30% from the total project effort

☐ 40% from the total project effort

☐ 50% or greater from the total project effort

37c. How deep on understanding the problem domain would you say you get during a software development project?

Very superficial **1 2 3 4 5** ***Domain expert***

38. Is there anything you would like to change in order to improve the current practice of developing shared understanding?

____

____

A.4. Closing Protocol

- **Asking for client access opportunity**: Is it possible for you to recommend us to one of your client for asking these questions and get their perspective?
- **Interview closing statement**.

Appendix B – Participant information sheet

Participant Information Sheet

Date Information Sheet Produced:

24 August 2007

Project Title

Tool and Technique Support for Cooperative Requirements Understanding

An Invitation

Both research and practice suggest that requirements engineering is an important but error-prone aspect of software engineering. Why is this and how can it be improved in practice? I invite you to participate in a research project that should help to answer these questions and contribute to improvements in practice and tools in this area.

My name is Christian Harsana. I am a student from AUT University, currently doing a research thesis as partial fulfilment of a Master of Computer and Information Sciences degree. The research for this thesis is in the area of requirements engineering. In particular it relates to understanding what tools and techniques are currently used in practice to support sharing of domain understanding in a software development project.

Since this research involves understanding current practice, an essential element is partnering with organisations that currently practice in this area of software development. This invitation provides your organisation with the opportunity to share your experience and perspectives and contribute to the body of knowledge in this area.

It is very important to note that your participation in this research is voluntary in nature, and you may withdraw your participation at any time without any adverse consequences.

The following questions and answers are intended to address the most common questions that the participant may ask about this particular research project. If you need further information, feel free to contact the researcher, Christian Harsana. My contact details can be found at the end of this document. It is recommended that you use e-mail to reach me.

What is the purpose of this research?

The purpose of this research is to investigate and better understand current practice of domain knowledge sharing between stakeholders in requirements engineering activities. The use of techniques and tools used in practice to support these activities, as well as perceptions of their efficacy, are the primary focus of the research. This is an important aspect of requirements engineering practice and theory that hasn't been given much attention in research literature. Understanding current practice and perceptions will suggest possible process improvements and lead to the design of better support tools. This will ultimately contribute to faster and more accurate software development.

At the end of this research a report summarising the main results will be made available to you if requested. Furthermore, it is expected that some papers will be published in academic journals relating to this particular research project, with all information kept anonymous.

How was I chosen for this invitation?

Your organisation has been selected because your main business is IT systems development, you are Auckland based and may have indicated an interest in partnering in research to improve software engineering.

research to improve software engineering.

What will happen in this research?

The project will consist of two phases of interview. In the first phase, participants from the partner organizations will be invited to share their experiences and perspectives regarding requirements engineering activities, focusing on domain knowledge sharing. Participants will be asked about the tools and techniques used in the process, and their perceptions on the effectiveness of them.

A second phase of interviews may be conducted to probe deeper into specific issues discovered during the first phase.

The interviews will take place on a one-to-one basis with the researcher. They will be held at the participant's usual work places or any neutral place to guarantee the comfort during the interview session.

What are the discomforts and risks?

During the interview session there are possibilities that participants might feel uncomfortable about sharing their experience and perspectives related to their own performance or their relationships with their colleagues. This kind of information might potentially put the participant in a risky situation if the interview is overheard.

Since the interview session will be recorded, some of the information (e.g. demographic data, etc.) or participant's voice could potentially reveal the participants' identities. Another potential risk is during the transcription of the interview record where the transcriber who does not have any importance with the research project may hear and read the interview results.

How will these discomforts and risks be alleviated?

In order to alleviate the discomforts and risk, participants will have an option for doing the interview at a neutral place beside their own usual workplace. It is believed that this can make the participant more comfortable in sharing their experiences and perspectives.

To address the transcriber issue the transcriber will sign a confidentiality contract, before the transcription process starts, that will oblige them to keep the interview record confidential.

Demographic data and the tape will be coded and the information about the code will be separated and stored separately, ensuring the protection of the participants' identities.

What are the benefits?

This research is part of a larger research project at AUT University related to improving the understanding of requirements engineering practice and theory and the improvement of practices and tools. As well as adding to the body of knowledge and influencing practice in this general area, the information will be made available to yourself and your colleagues and it is hoped that the knowledge gained will be useful for improving the practice in your organization.

How will my privacy be protected?

All of the materials related to the participants' information (consent form, tape, interview transcript, and interview note) will be stored at AUT in a locked drawer for at least 6 years. After that the material will be destroyed.

If you decide to withdraw from this research project for any reason it is guaranteed that all of the materials relating to you will be destroyed as soon as practicable after your request.

In addition, your employer will not have access to any interview records except when it is required under New Zealand law.

What are the costs of participating in this research?

The interview will take between 1 to 1.5 hours of your time. A similar time may be required at a different phase if you elect to participate in phase two also.

What opportunity do I have to consider this invitation?

Due to time restrictions in undertaking the field work for the research, we would ideally like to have notice of your agreement within a week of you receiving this invitation.

How do I agree to participate in this research?

To follow up on this invitation to participate in this research, please complete and sign the attached Participant Consent form and return using the self-addressed envelopes. Where appropriate please obtain the consent of your employer before agreeing to participate.

Will I receive feedback on the results of this research?

If you would like a report summarising the results of this research, please tick the appropriate box on the consent form.

What do I do if I have concerns about this research?

Any concerns regarding the nature of this project should be notified in the first instance to the Project Supervisor,

Jim Buchan
Senior Lecturer
School of Computing and Mathematical Sciences
Auckland University of Technology
Private Bag 92006
Auckland 1142
New Zealand
Phone: + 64 9 921 9999 x 5455
Email jim.buchan@aut.ac.nz

Concerns regarding the conduct of the research should be notified to the Executive Secretary, AUTEC, Madeline Banda, *madeline.banda@aut.ac.nz* , 921 9999 ext 8044.

Whom do I contact for further information about this research?

Researcher Contact Details:

Christian Harsana Ekadharmawan
Master of Computer and Information Science Research Student's Lab,
School of Computing and Mathematical Sciences
Auckland University of Technology
Private Bag 92006
Auckland 1142
New Zealand
Phone: + 64 9 921 9999 x 5002
Email Christian.harsana.aut.study@gmail.com or mvt6204@aut.ac.nz

Project Supervisor Contact Details:

Jim Buchan
Senior Lecturer
School of Computing and Mathematical Sciences
Auckland University of Technology
Private Bag 92006
Auckland 1142
New Zealand
Phone: + 64 9 921 9999 x 5455
Email jim.buchan@aut.ac.nz

Approved by the Auckland University of Technology Ethics Committee on *28 September 2007*

AUTEC Reference number *07/170*

Appendix C – Participant consent form

Consent Form

For use when interviews are involved.

Project title: ***xxx***

Project Supervisor: ***xxx***

Researcher: ***xxx***

○ I have read and understood the information provided about this research project in the Information Sheet dated dd mmmm yyyy.

○ I have had an opportunity to ask questions and to have them answered.

○ I understand that notes will be taken during the interviews and that they will also be audio-taped and transcribed.

○ I understand that I may withdraw myself or any information that I have provided for this project at any time prior to completion of data collection, without being disadvantaged in any way.

○ If I withdraw, I understand that all relevant information including tapes and transcripts, or parts thereof, will be destroyed.

○ I agree to take part in this research.

○ I wish to receive a copy of the report from the research (please tick one): Yes○ No○

Participant's signature: ....
Participant's name: ....
Participant's Contact Details (if appropriate):
……………………………………………………………………………………..
……………………………………………………………………………………..
……………………………………………………………………………………..
……………………………………………………………………………………..
Date:

Approved by the Auckland University of Technology Ethics Committee on** **type the date on which the final approval was granted** **AUTEC Reference number** **type the AUTEC reference number

Note: The Participant should retain a copy of this form.

Appendix D – Participant invitation letter

To
IT Manager of
….
….
….

18 October 2007

Re: Techniques used to support understanding of user requirements.

Dear Sir/Madam,

It is widely acknowledged that the misunderstanding of user requirements is one of the main contributors to unsuccessful or challenged software projects. This is because requirements management is typically a very complex and error-prone process. A number of practices and tools have evolved to support and improve this process. But do these "best practices" provide tangible improvements in software quality? Are these practices actually used by successful New Zealand software companies?

It is exactly these types of questions that we are addressing in some research at AUT University, with an aim to contributing to best practice in this area.

The sharing of understanding about business problems and goals between developers and users is one important aspect of cooperative development of user requirements. This often involves collaboratively refining understanding of the problem or opportunity as a project progresses and frequent confirmation of common understanding. It is a pragmatic understanding of the techniques and tools that are used *in practice* and how these overcome barriers to shared understanding that is the subject of this research project. This is why we need the involvement of important software development industry players such as your organisation as research partners.

Your organisation is invited to participate in this research.

We invite a project manager, analyst, developer, or other appropriate representative from your organisation to contribute to this research by taking part in some brief interviews, as described in the documentation accompanying this letter. In return you will be provided with a summary of the results of the research and may better understand how requirements management practices could provide further business value to your software development processes. Confidentiality will be paramount throughout this process and the research project has passed the scrutiny of the University's ethics committee.

This research is being conducted by a research team from AUT comprising Jim Buchan, Professor Stephen MacDonell, and Christian Harsana. Christian is a Masters student undertaking his thesis and has some experience working in the IT industry and is aware of the practical challenges project managers and IT organisations face. Stephen MacDonell is Professor of Software Engineering and Director of the Software Engineering Research Lab (SERL) at AUT. He has been researching topics in software engineering management since 1988. Jim Buchan is a senior lecturer at AUT in software engineering as well as having over 12 years extensive practical IT and business experience.

To confirm your willingness to contribute to this project, or if you would like more information, please contact Christian Harsana either by email at **christian.harsana.aut.study@gmail.com** or by posting us your business card in the return envelope received with this letter. You will then be contacted in a few days regarding the next step in partnering with us on this research project.

We trust you will see the value in this investigation and look forward to your reply.

Yours sincerely,

Christian Harsana, Jim Buchan and Professor Stephen MacDonell,
School of Computing and Mathematical Sciences,
AUT University.