University of Fort Hare

Together in Excellence

A MODEL FOR USER REQUIREMENTS ELICITATION SPECIFIC TO USERS IN RURAL AREAS

by

NAOMI ISABIRYE

A MODEL FOR USER REQUIREMENTS ELICITATION SPECIFIC TO USERS IN RURAL AREAS

by

Naomi Isabirye

Dissertation

Submitted in fulfilment of the requirements for the degree

Masters of Commerce

in

Information Systems

in the

Faculty of Management and Commerce

of the

University of Fort Hare

Supervisors: **Prof. Stephen Flowerday and Prof. H.N Muyingi**

January 2009

Abstract

Dwesa is a rural town situated in South Africa's Eastern Cape Province that has been selected as a testing site for an e-commerce project to address some of the challenges faced by the community. These challenges include difficulties stemming from poverty, poor access to public services, unemployment and low levels of literacy. The Siyakhula Living Lab project's aim is to develop an e-commerce platform that will 'connect' the residents with the necessary access to services and markets by providing them with the necessary tools that can help in alleviating some of their challenges. In order for the project to achieve its objectives, a deeper understanding into the needs of the prospective users is required. User requirements elicitation deals with the process of interacting with the prospective users to understand and document their needs. This research aims at evaluating existing requirements elicitation techniques and methodologies in the context of rural information technology implementations. The primary objective is to develop a model for user requirements elicitation in Dwesa. Many requirements elicitation techniques and frameworks exist, but few have been evaluated in the context of rural software implementations. Requirements elicitation techniques should not be applied simply as steps to gather information. Instead this research project proposes a model that can be applied to assimilate the contribution of knowledge regarding the stakeholders, problem and solution characteristics, and other characteristics into the software development process for the effective elicitation of requirements.

Declaration

I _____, hereby declare that:

- The work in this dissertation is my own work.
- All sources used or referred to have been documented and recognised.
- This dissertation has not previously been submitted in full or partial fulfilment of the requirements for an equivalent or higher qualification at any other recognised educational institution.

_____

Table of Contents

Table of Figures

List of Tables

Acknowledgements

This research is the result of the support, effort, and sacrifice of a group of people to whom I am sincerely grateful:

1. My two supervisors, Prof H.N Muyingi, who introduced me to the project and provided ongoing support at all levels throughout the duration of the project and Prof S. Flowerday, who offered persistent guidance and support to ensure that the project was completed.

2. My parents, my husband Gus, my siblings (Charles Tony and Esther) and my new extended family members (Pinky and Eva), as well as my closest friends, for their unconditional and unwavering support and encouragement through the process, and my son Anzo for being my constant inspiration.

3. The residents of Dwesa, specifically those involved in arts and crafts, and the teachers at Mpume Primary School who helped in facilitating the interviews.

4. The software developers of the Siyakhula Living Labs Project for the efficiency and eagerness they displayed in their participation in the survey, as well as the rest of my peers in the Siyakhula Living Labs Project.

5. Finally to the bodies that made this research possible by backing me financially, the Telkom Centre of Excellence for Developmental E-Commerce of Fort Hare and Rhodes Universities and its respective sponsors, The National Research Foundation and the Govan Mbeki Research and Development Centre of Fort Hare University.

In loving memory of the late Prof. Rinette Roets, my initial research supervisor whose kindness, encouragement and support, matched her academic contribution to this research and is sincerely appreciated.

Part 1: Background and Introduction

Chapter 1

Introduction

Dwesa is a rural town located along the Eastern Cape coast hosting a nature reserve with accommodation facilities for holiday makers. Dwesa has a rich cultural and political heritage that has resulted in it being the subject of extensive social research. Research conducted by Fay, Timmermans and Palmer (2002) reveals that Dwesa's ecological diversity has caused it to be a highly protected area in terms of conservation policies. Unfortunately, as explained by Fay, Timmermans and Palmer (2002), the consequences of historical injustices and innumerable conservation laws surrounding the use of natural resources in Dwesa have grossly incapacitated the residents. In 1994, revolts by the Dwesa residents attracted extensive media attention and by 2001, Dwesa became one of the first communities in South Africa to successfully reclaim ownership of their land. Over a decade later, conservation policies still restrict the use of resources, and residents are still critically impoverished.

Aside from the conservation restrictions, entrepreneurs in Dwesa are also restricted by poor access roads and lack of telecommunications infrastructure, electricity and basic public services. Entrepreneurs are therefore limited to marketing their goods or services over a small geographical region, and those that target the tourist market are subject to seasonal income. It is believed that an e-commerce platform will expand the boundaries of their market, and hence, boost the economic activity within their society.

A research project conducted by the Telkom Centres of Excellence for developmental e-commerce under the universities of Fort Hare and Rhodes is currently underway to implement an e-commerce platform in Dwesa. One of the objectives of the project is to develop an e-mall from which entrepreneurs in Dwesa can trade their arts and crafts. To enable this, the project is engaging in research on the provision of connectivity, training, infrastructure and software. In order to ensure that the software fulfills the needs of its prospective users, an effective process of requirements elicitation is imperative. The aim of this research project is to uncover the most effective approach to eliciting the user requirements that are necessary for the effective adoption and use of the e-commerce platform by entrepreneurs in Dwesa. The difficulty in applying traditional elicitation techniques and methodologies is that many of them have been developed and tested in contexts

vastly contrasting Dwesa's. Given the socio-economic and cultural atmosphere, they may not be effective in identifying the real needs of e-commerce users in rural areas. This study will evaluate existing methodologies surrounding the elicitation of user requirements with the aim of contextualising them to rural environments.

1.1 Statement of Problem

i. How can the user requirements of e-commerce applications for rural South African entrepreneurs be effectively elicited in order to define the needs of the users in order to promote the adoption and sustained use of the applications?

Amongst the many challenges in implementing an E-Commerce platform in rural Dwesa lies the fact that, being situated in a remote and underdeveloped part of the Eastern Cape, many residents have had little or no exposure to computers. Furthermore, Dwesa residents and the systems developers come from vastly contrasting backgrounds and are divided by language, culture and education. For these reasons, it is important that a high level of interaction is maintained between developers, potential users of the system and members of the community within which it will be used. It is also necessary that a firm understanding of the human factors (for example social, cultural and literacy) is gained in order to identify the real needs of the community, and hence establish whether an E-commerce platform can in fact address their needs. This will allow analysts to pre-empt any possible causes of failure or resistance surrounding the use of the system and encourage ownership of the system by the community.

One of the measures that must be addressed in order to achieve the sustained use of the system is that the user requirements are adequately captured. Many techniques, guidelines and frameworks have been developed to support the elicitation of user requirements, many of which are centred on a well understood group of users. From a software development perspective, Dwesa's potential users are in the category of the less understood (non Western) users. Existing elicitation techniques should therefore be considered critically when being applied to this distinct category of users. This is likely to result in the adaptation of existing elicitation techniques or the introduction of new techniques to address the limitations of existing techniques in the given context.

1.2 Statement of Sub Problems

i. What are the specific characteristics that differentiate the use of ICTs in rural areas from typical urban ICT implementations?

This section aims at distinguishing the use of ICTs in rural areas from its use in urban areas. It is believed that the needs of the community will largely influence the adoption of the new technology. Dwesa is located in a very remote area. The telecommunications infrastructure is inadequate, and there is limited access to basic public services. Technology that addresses these needs will be eagerly adopted by the community. Additionally, it is recognised that the levels of literacy, cultural values and previous exposure to technology could have an effect on the use of ICTs in Dwesa.

ii. How do factors surrounding users in Dwesa challenge the use of traditional elicitation methods?

By distinguishing the use of ICTs in rural areas from its use in urban areas, it can be deduced that the software development process should be adapted to the context within which the system is developed. This calls for existing techniques to be reviewed and contextualised to ensure that the real needs of the users are understood and addressed.

iii. How can traditional elicitation methodologies be adapted to accommodate the distinct needs of users in Dwesa?

This section aims at critically evaluating existing requirements elicitation methodologies, with the purpose of adopting or adapting them for use in the context of Dwesa's E-commerce platform.

iv. How can the user requirements elicitation process be modelled to ensure that it uncovers the real needs of the prospective users effectively?

Based on the evaluation of existing requirements elicitation methodologies, an approach to requirements elicitation can be adapted to guide the development of ICT for rural users. This section aims at devising a methodology for requirements elicitation that draws on the strengths of various requirements elicitation methodologies.

1.3 Objective of Study

The objective of the study is to devise a requirements elicitation methodology that is suitable for users of e-commerce in rural areas that can be adopted by software development projects in similar settings. Thus, a system is more likely to be used sustainably when it addresses the needs of its users.

1.4 Literature Review

1.4.1 Characteristics Differentiating Rural ICT Users from Typical Urban Users

South Africa, along with most developing countries is challenged with the need to overcome the broad divide in capabilities and access to technology of its population. The type of needs facing rural communities distinguishes the implementation of ICTs in rural areas profoundly from urban areas. In Dwesa where the problems faced by the community include, according to Fay and Palmer (2002), poor road networks; inadequate health care, sanitation and education facilities; and an absence of electrical and telecommunications infrastructure, ICT implementation may not be considered to be a priority.

Hugo (2002) states that achieving socio-economic equity requires that emphasis is placed on education, training and resource accessibility. Unfortunately the developing world, specifically Africa, faces an enormous skills deficit, and potential users of ICT services are often widely dispersed with a significant proportion based in rural areas, where the skills shortage is obviously

more profound. According to Hugo (2002), the reality faced by urban South Africans as opposed to rural South Africans is very distinct. Hugo (2002) explains that while middle to high income urban South African communities have eagerly adopted ICTs and E-Commerce is widely accessible and even essential amongst the privileged, many Africans have had little or no exposure to even the most basic Information Technologies. This reality is explicitly exhibited in Dwesa where up until recently, there has been no electricity, telecommunications infrastructure let alone computers for use by the ordinary residents.

The use of ICTs often requires a basic ability to communicate in English. Unfortunately, the majority of Dwesa residents either cannot speak English at all or speak it at basic level. Either way English serves as a second language to the indigenous community at large. This pronounces the challenge faced by analysts in extracting the user requirements of proposed IT systems. It also presents the necessity to undertake measures to address the language issues in the user interface.

1.4.2 Factors Surrounding Rural Users That Challenge the Use of Traditional Elicitation Methods

According to IEEE (1990), a requirement can be defined as follows:

1. A condition or capability needed by a user to solve a problem or to achieve a goal
2. A condition or capability that must be met or possessed by a system component to satisfy a contract, standard, specification or other formally imposed documents
3. A documented representation of a condition or capability as in 1 or 2

Requirements elicitation can therefore be defined as the process of uncovering and documenting the capabilities required by users to solve a set of problems or achieve a set of goals. There is a common misconception that requirements are made up of functions. According to Christel and Kang (1992) this perception is propagated by structured analysis techniques that often overemphasise the definition of functional requirements. Southwell (1987) in Christel and Kang (1992) categorises requirements as functional and non functional, where he classifies performance, reliability, interfaces and design constraints as non functional requirements. In Ashworth's (1989) definition in Christel and Kang (1992), he emphasises functionality, data, goals, design constraints and implementation constraints as categories of requirements. The author supports Christel and

Kang's criticism of the above classical definitions of requirements as they all focus on the technical goals, and place little emphasis on the users' needs.

Given the cultural differences between analysts and users in rural ICT implementations, there is a need to develop a deeper understanding of the problem domain, by first evaluating the human factors surrounding users before selecting a method of interacting with them to identify or define their needs. This is critical as any errors made at the beginning of the requirements phase will have a lasting impact on the project as a whole. For instance, if the developers are perceived to be disrespectful or untrustworthy by the users, they will not engage or cooperate with them throughout the development process.

According to Disse (2001) system requirements are best determined through merging the different viewpoints of the stakeholders involved. Disse argues that in a multicultural system design setting, the forming and interpretation of viewpoints is problematic because it depends on the stakeholders' perception of the environment. Yet the perception of reality and the structuring and processing of experience depends on habits that are shaped by culture. The factors challenging the use of ICTs in rural areas add complexity to the challenge of determining the user requirements of systems in rural areas because traditional elicitation techniques may not always be suitable for the target users.

Heeks (2002) believes that gaps are likely to exist between the system design which is based on the analyst's perceptions, and the reality, which is often foreign to the analyst. Heeks (2002) explains that gaps are a result of the detachment, in more than just physical terms, of dominant design stakeholders from the context in which the proposed system is to be implemented. In order for the ICT system to adequately suit its environment, it is essential for the developers of the system to understand the context within which it is implemented. The occurrence of design-reality gaps means that the project does not suit the context within which it is to be implemented.

1.4.3 Adapting Traditional Requirememnts Elicitation Methodologies to Accommodate the Distinct Needs of Rural Users

Successful requirements elicitation is very critical to the overall success of any software development project. Unfortunately its application in rural ICT implementations has been largely neglected in research. Browne and Rogich's Task Characteristics Model (2001), Davies and Hickey's Techniques Selection model (2004) and Christel and Kang's Requirements Elicitation Process Model (1992) have been identified by the author as models with a strong inclination to be adapted to a rural context. Both the Task Characteristics Model and the Techniques selection Model's recognise the relationship between the problem and solution domains, the stakeholders, the project/organisation, and the elicitation process as their key strengths. The main difference between the models is that Browne and Rogich (2001) prescribed the use of substantive and procedural prompts to interface the relationship, while Davies and Hickey (2002) established a technique selection process to determine how to interface the relationship. The main shortcomings of the models is the rigidity of their prescriptions of the techniques that should be applied, as well as the diagnosis of variables that determine the technique selection process.

In Contrast, Christel and Kang's (1992) model neither prescribes the use of specific techniques, nor attempts to identify the variables affecting technique selection. The main emphasis of the model is the effective management of the requirements and the relationship between analysts and users throughout the software development process. Unfortunately, the model's failure is that it does not adequately elaborate on how the initial requirements are obtained.

Zhang's (2007) Matrix of Technique Selection does not prescribe specific elicitation techniques to apply. It does however identify the factors that influence the technique selection process, but expresses these factors as categories rather than individual variables. Zhang's (2007) approach is more practical, flexible and specific as it guides the analyst in terms of the manner in which various techniques address various issues, without dictating the techniques to apply

A model that consolidates the strengths of these models, while minimising or eliminating their shortcomings, would be ideal for the elicitation of user requirements in a rural contexts.

1.4.4 Modelling the User Requirements Elicitation Process to Ensure the Effective Determination of the Real Needs of the Users

Heeks (2002) points out that there is often a significant amount of subjectivity involved in assessing the failure or success of a project. One view is that success should be measured in the light of the goals of the project funders and implementers, while others argue that it should be measured from the perspective of the participants. One of the important outcomes of this research is the establishment of a method of requirements elicitation that facilitates the achievement of the goals of both the users as well as the implementers of the system. Measuring the success or failure of the development project is difficult as the goals of the community and the goals of the developers are not always aligned. For instance, Schneiderman (1998) recognises that differences exist between the goals of system engineers and user interface designers. He states that system engineers strive towards achieving goals such as adequate functionality, reliability, availability, security, data integrity, standardisation, integration, consistency and portability. User interface designers, however, focus on issues such as learning curves, speed of performance, rate of errors, retention over time and subjective satisfaction. Myers (1997) also acknowledges how acceptance and use of a new technology are affected by goals and expectations. He states that technology use and acceptance will be directly correlated with perceptions of net benefit by those who use it, and emphasises the fact that benefit to an organisation is not the same as benefit to users. Cooper (1997) reaffirms this view in an article regarding goal directed design, where he emphasises the differences between user goals, developer goals, corporate goals and practical goals. He explains that the corporate goals of an organisation's management would be based on profitability, whereas the goals of the user would be to avoid work. The ideal system in this case would be one where the system makes the users work easier, and results in profit maximisation. This study will attempt to apply the principle of Christel and Kang's (1992) model as a means of facilitating communication between the analysts and the users to ensure that the goals of all the stakeholders are addressed throughout the projects.

When developing a system for a community that has no framework of comparison in terms of performance standards and technical specifications, many software requirements specifications methodologies are inadequate as they are designed to verify the requirements of stakeholders who already know what to expect from a system. An effective mechanism for eliciting, modeling and

verifying the needs of the user is critical as this will serve as a means of ensuring that an alignment between the goals of users and systems developers is maintained. Zhang (2007) recommends the application of synthetic requirements elicitation techniques to achieve this as they serve to merge the perceptions of the users and analysts.

1.5 Research Methodology

This research will use an interpretivist approach to evaluate conditions in Dwesa. According to O'Brien (2001), interpretivism is characterised by a belief in socially constructed, subjectively based reality that is influenced by culture and history. Within the interpretivist paradigm, the research will engage a Participatory Action Research (PAR) approach to answer the research problems and will engage the principles of Participatory Design in the requirements analysis activities. Action research is described by Lewin (1946) in O'Brien (2001) as "a comparative research on the conditions and effects of various forms of social action". According to Lewin (1946), action research involves a spiral of steps, each composed of planning, action, and fact finding about the result of the action. This follows the belief that there is an abundance of knowledge that can be drawn from the community, and likewise, the community can gain profoundly from the knowledge of the researchers. The synergy between the two diverse spheres of knowledge will be of great value to the project as a whole, but can only be derived through continuous interaction and collaboration between the researchers and the community.

According to Mouton (2001) PAR involves the subjects of the research as an integral part of the design and usually engages the use of qualitative methods to gain an insight into the lives of the research subjects. Mouton (2001) adds that PAR is often associated with an explicit commitment to the empowerment of the participants and is likely to result in a high level of involvement by the participants and hence enhance the chances of high construct validity, low refusal rates and high ownership of the findings. The main weakness however, according to Mouton (2001), is the ability of the researchers to affect the research process to serve their own interests.

1.6 Delimitation of Study

This paper will focus primarily on the E-commerce initiative based in the rural community of Dwesa. It is expected however that the results will be generalisable for other South African rural ICT Implementations. This research acknowledges the likelihood of vast differences existing between rural entrepreneurs in developed countries and rural entrepreneurs in developing countries, therefore, although literature from the experiences of rural areas in developed countries will be reviewed, the case study will be conducted in reference to rural areas in less developed countries (LDCs). The report will be limited to the requirements management aspect of the development of the Siyakhula Living Labs project in Dwesa.

1.7 Chapter Outline

Chapter 1: Introduction

Chapter 2: Differentiating the Use of ICTs in Rural Areas from Its Use in Urban Areas

Chapter 3: Contextualising User Requirements Elicitation Techniques to Users in Rural Areas

Chapter 4: A Review of User Requirements Elicitation Methodologies

Chapter 5: A model for Requirements Elicitaition Specific to Users in Rural Areas

Chapter 6: Research Design

Chapter 7: The Siyakhula Living Labs Project: A Case Study

Chapter 8: Conclusions

Part 2: Literature Review

Chapter 2

Differentiating the Use of ICTs in Rural Areas from Its Use in Urban Areas

2.1 Introduction

The digital divide is amongst the most widely discussed issues with regard to technology for development. Impoverished communities stand to gain from the ability of ICTs to leverage information and knowledge for the improvement of the economic, social and physical conditions they face. Unfortunately, the challenge of implementing ICTs in rural areas is burdened by the fact that the use of software often requires an ability to communicate in English or one of the few European or Asian languages catered for in the development of widely used software. Furthermore, the lack of electric or telecommunication infrastructure in rural areas makes any initiatives to start ICT projects very costly. Initiating ICT projects in rural areas is also challenged by the fact that rural communities struggle to meet their most basic needs and participating in ICT projects may not always fall amongst their priorities.

The aim of this chapter is to identify the issues surrounding rural communities that affect ICT implementations in rural areas. The chapter explores matters surrounding the digital divide, language, the high cost of technology, social issues, and information as an empowering resource for rural communities.

2.2 The Digital Divide

The digital divide is a worldwide phenomenon but its consequences are more adverse in developing countries as a higher proportion of the population is underprivileged and the impact is therefore more material. South Africa faces a huge challenge in addressing the divide in access to technology of its population. According to Hugo (2002), achieving socio-economic equity requires that emphasis be placed on education, training and resource accessibility. Unfortunately, according to Hugo (2002), the developing world is plagued by an enormous skills deficit, and users or potential users of IT services are often widely dispersed with a significant proportion based in rural areas. In a Rural development research paper delivered by Hemson, Meyer and Maphunye (2004), it is stated that South Africa's non-urban population amounts to 45% of the total population and of these, 85% live in the impoverished former homelands. According to the International Telecommunications

Union (2003), only 1.3 computers existed per 100 Africans. Although the figures for South Africa were better than for Africa as a whole (7.26 computers and 7.1 internet users per 100 South Africans), they did not reflect the proportion of users that were in rural areas. Hugo (2002) states that many South Africans are unaware of the reality faced by the low income proportion of their society. He explains that the adoption of ICTs amongst middle to high income South Africans continues to rise while many of the poorer communities have had little or no exposure to even the most basic Information Technologies. The high adoption of new technologies amongst South Africa's privileged is thus likely to paint a deceptive picture of the actual situation. In Dwesa, where electricity and telecommunications infrastructure have just recently been introduced, and few residents have had any contact with computers, the divide in access to ICTs is vividly evident.

2.3 The High Cost Of Technology

The high cost of ICTs services is amongst the major factors contributing to the disparities between the availability of ICTs in developing countries from developed countries. Petrazzini and Kibati (1999) distinguish between the costs by separating them into two cost categories, setup costs and operating costs. Setup costs are usually high regardless of where the infrastructure is being deployed. However, setup costs relative to per capita income are much higher in developing countries than they are in developed countries. In the United States of America (U.S.A), the average professional can comfortably afford three computers for the equivalent of a month's salary. In Tanzania however, a computer costs three times an average professional's monthly salary (Petrazzini and Kibati, 1999). Moreover, the professional work force is much smaller and the ratio of professional workers to the total working population is much lower in developing countries therefore the amount of internet development in most of these countries cannot go beyond the small professional class. Operating costs in developing countries are also high. Although the absolute price difference might not be significant, the purchasing power of most people in the developing world makes access to Internet services extremely expensive.

Petrazzini and Kibati (1999) identify the differences in market characteristics between countries as a major underlying reason for the great disparity in internet access prices. They support this theory with the fact that, in most low-income countries, the most prevalent market structure is one in which

the Internet Service Provider (ISP) market is open to competition, while basic telephone services are still under monopoly conditions. Given the challenges in implementing ICTs in developing countries, the task of doing so in the rural parts of these countries could be daunting. Failure to set up the necessary infrastructure in the past has resulted in rural communities being left behind their urban counterparts. If initiatives are not made to address the inequities between urban and rural areas in terms of infrastructure, the digital divide will continually grow broader.

2.4 Language Issues in ICTs

Technology in rural parts of developing countries is subject to complex cultural influences. Computer users in rural communities are from backgrounds profoundly different from those for whom the technology was originally developed. Issues surrounding language and low levels of Computer literacy in rural areas, in comparison to urban areas, pose further strain on implementation of new technologies.

One of the major challenges concerning the adoption and use of technology in multicultural societies lies in the possibility of conflicting cultural values and beliefs, needs, and lifestyles in the various cultural groups. The user requirements are likely to differ depending on the context within which the technology is being used. It is therefore necessary to determine the degree to which multiculturalism affects computer users in their specific context, before addressing the design of the interfaces with which they interact. The use of ICTs often requires that the users can communicate in English. Unfortunately, the majority of Dwesa residents either cannot speak English at all or speak it at basic level. Research into localisation of the user interface to accommodate language barriers is therefore vital.

Information Technologies and Education for the Poor in Africa (ITEPA, 2004), state that although there is a recognised importance of literacy in indigenous languages. They argue that the use of international languages in education is more practical and cost-effective and ascribe problems associated with education in indigenous languages to the following factors:

- Inadequate materials in local languages
- Lack of research-based materials in local languages

- Poorly trained teachers in local languages

This perspective is supported by Blignaut, Burger and De Wet (2002), who conducted a study at the University of The Free State that found that most students that use an African language as a first language can search for information online in English more effectively than in their mother tongues. Afrikaans students that participated in the survey, however, were more comfortable using an interface that had been translated into Afrikaans. They attribute this to the fact that in South Africa, historically, the academic material has been available predominantly in English and Afrikaans, and furthermore, many scientific terms cannot be translated directly into other indigenous languages. Their observation is a possible demonstration of the ability of individuals to adapt according to their circumstances. The students that have been exposed to English academic material, have adapted, and evolved their mental models to a level whereby they can grasp web content faster in English than in their mother tongues. Many Afrikaans students on the other hand have been schooled in their mother tongue, and can easily comprehend web material that has been published in a familiar language (Afrikaans).

The main shortcoming of this observation is that university students do not form a representative sample of South Africa's population, especially since 80% of Black South Africans, and 40% of White South Africans cannot read beyond a grade seven level (Hugo, 2000). University students form the smaller portion of the population that have been privileged enough to have reached a tertiary education level, and have had at least 12 years to adjust to English or Afrikaans as languages of instruction.

The study does however suggest that customisation of interfaces depends largely on the category of users that are being targeted by the system. Cooper (1997) uses the example of software in a jet fighter cockpit, where the users are highly trained pilots. He suggests learnability should not be a primary design goal for the software, because a pilot who has been trained to learn how to use highly specialised weapons systems should not have much difficulty learning how to use new software. Instead, responsiveness should be a primary concern as it will be critical in combat situations.

Many studies have emphasised the need for adaptation to local conditions. Lachman, Nedd and Hinings (1994), highlight the importance of adapting information practices and IT implementations to the core competencies and values of a nation. Norton (2003), however, believes that learning how to use standard interfaces yields a higher value than customisation, and organisations should focus on rapidly advancing technology skills, and emphasise training rather than customisation as a solution to bridging the gap.

Norton (2003) suggests that the following steps be taken when designing systems for developing countries to address the varying mental models amongst individuals:

1. Evaluate the situation from a broad perspective.
2. Engage in all necessary measures to ensure that training is provided.
3. Understand the general level of previous exposure to technology of a specific user group.
4. Build on previous exposure to take advantage of the mental models that users already have.
5. Identify the gap between existing mental models and the knowledge and understanding required to use the technology for a specific task.
6. Decide how to bridge that gap by considering how much design intervention is required and the level of previous training and exposure to technology.

2.5 Social Problems

Maslow (1970) classifies the needs of human beings into five broad categories; physiological needs, safety and security needs, the need for love and belonging, esteem needs, and the need for self actualization. Maslow's model represents the order of priorities of human needs. Physiological needs such as food, water and sanitation must be satisfied first before addressing higher order needs. In Dwesa, the challenges confronting the residents, according to Fay and Palmer (2002), include poverty, unemployment, inadequate health care, sanitation and education facilities, poor road networks, and an absence of electrical and telecommunications infrastructure. According to Mwabu and Thorbecke (2001), rural areas are characterised by high levels of poverty associated with low agricultural productivity, environmental degradation, unsustainable population growth, poor rural infrastructure, inadequate access to markets and market information, low investment in human

resources, ethnic and tribal conflicts, the HIV/AIDS pandemic and high disease burdens, inappropriate economic policies and adverse effects of globalization amongst others.

ICT implementation may not be considered to be a priority when considering the fact that so many of the challenges faced by rural communities affect their primary needs. Furthermore, new technologies in urban areas are often developed to satisfy known needs such as improving efficiency within organisations, providing entertainment, or enabling knowledge, communication and information transfer. These technologies often filter into rural areas with little consideration into their ability to serve the needs of their rural users. Access to E-mailing facilities for instance, is of little benefit to someone who is not part of a community of people who depend on E-mails as a form of communication and have regular access to E-mailing facilities.

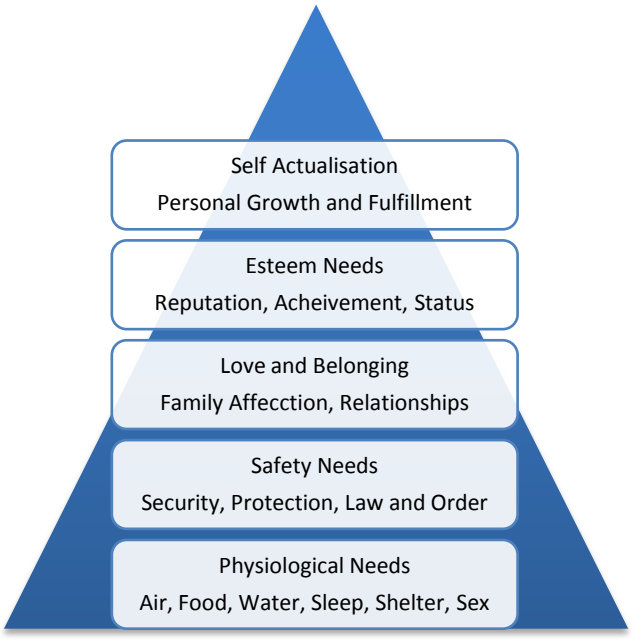


Figure 1: Maslow's Hierarchy of needs (Maslow(1970)

2.6 Information and Communication Technologies as an Empowering Resource

According to the International Institute for Sustainable Development (1995), Information is critical to the attainment of sustainable development. Zake (2007) describes insufficient and inaccurate

information and knowledge as one of the traps that keep rural communities in a cycle of poverty and results in consequent series of poor choices and actions. Zake describes information and knowledge as enablers that facilitate better actions and choices.

Often, too much emphasis is placed on the technological elements of ICTs, and not enough emphasis on the elements that the technology is meant to enable (information and communication). Availing computers, telephones and connectivity will not single handedly solve the problems of rural residents. Although attaining specific information and communication technologies may not be seen as a priority for poor rural communities, the ability to communicate and have access to relevant information is of paramount importance. The high penetration of mobile phones throughout urban and rural areas of developing countries is an illustration of the need for communication even in the poorest communities. Rural communities likewise, could benefit profoundly from increased access to information.

During one of the visits to Dwesa, the research group found that the computers that had been installed in some of the schools had not been in use for over a week. The reason for the redundancy was that following a few days of heavy rains, the electricity supply in the area had been disrupted due to damage to the infrastructure. The community members were unaware of the need to report the fact that there was no electricity, so they simply waited for it to return. When the project group arrived and was made aware of the electricity shortage, they made a few telephone calls, and were requested to collect certain information from the various powerlines before the problem could be resolved. After driving around and gathering the necessary information, and communicating it back to the electricity service providers, the electricity supply was soon restored. The experience depicted the manner in which access to information (regarding who to consult and how to reach them) and the availability of relevant communication tools could help the residents solve basic problems. Furthermore, the fact that the project group had access to transport, despite the poor state of the roads, allowed the situation to be resolved effectively. Dwesa residents are mainly pedestrians and not all parts of Dwesa have cell phone network coverage. The situation may have remained unresolved indefinitely.

The limited ability to communicate and access information in Dwesa has closed the community off to services that have the potential to improve their standard of living. Rural areas as a result are further excluded from developments implemented by their urban neighbors. ITEPA (2004) state that viewing ICTs in the light of a knowledge society which, in their view, is fundamentally changing most aspects of human activities, is a broader way of looking at the contribution of ICTs than merely perceiving the role of ICTs as a set of potential delivery and instructional tools.

According to the United Nations Development Program (UNDP, 2001) a country's ICT investments can directly enhance the capabilities of its citizens, increase skill capacity and in turn, support further development and increase the productive use of the technological infrastructure. The growing sophistication of the skill base and the technological infrastructure can lead to innovation and the creation of new knowledge and new industries.

Practical applications of ICTs to support service delivery to rural residents are provided by Zake (2007). These include providing information about human and/or legal rights, the provision of basic services and how to access them, birth and death registrations, social welfare information, diseases associated with sanitation, subsistence farming relevant to the specific area and infectious diseases and epidemics.

2.7 Conclusion

The high cost of technology relative to the per capita income, language and literacy issues, lack of infrastructure and general social challenges that pervade rural communities make the effort to implement ICTs in their communities more difficult than in urban areas. The inability of rural communities to access relevant information hampers their ability to make decisions and undertake actions that can improve their situation.

The inadequate infrastructure (civil, electrical and telecommunications) is among the primary reasons that the inequities have not been adequately addressed. They form the foundations upon which any further developments can be implemented and sustained. Unfortunately, the primary needs of the community may be more of a priority and their enormity may be the reason rural areas

have remained on the negative side of the digital divide. This chapter revealed the challenges affecting rural ICT implementations. The chapter that follows will review a variety of requirements elicitation techniques in light of these challenges.

Chapter 3

Contextualising User Requirements Techniques to Users in Rural Areas

3.1 Introduction

Three visits to Dwesa were undertaken, each lasting one week, to engage extensively with entrepreneurs in Dwesa to investigate the challenges in eliciting user requirements for the E-Commerce platform that is in the process of being deployed there. The first trip was aimed at getting acquainted with the community, while the subsequent trips were targeted at identifying entrepreneurs to participate in the project, introducing them to the project and demonstrating a prototype of the proposed system. The prototype exhibited the basic functionality of any standard e-commerce website, and served the purpose of demonstrating the concept of E-Commerce to the users, and identifying aspects of the system that the users did not understand. The interaction with the entrepreneurs provided more insight as to the effectiveness of various requirements elicitation techniques.

Although interviews proved to be the most practical and effective means of gaining a broad understanding of the needs of the community, their main limitation was that they only provided answers to the questions asked, and hence left room for unexpected factors being overlooked. Furthermore, it had not been anticipated that failure to communicate the goals of the interview to the translator could negatively impact the exercise by disrupting the message between the interviewer, translator and interviewees.

Group elicitation techniques, according to Silver and Wood (1995) produce more and better solutions to problems, utilise resources efficiently, eliminate inferior ideas more effectively, increase productivity, result in superior judgements, have a faster learning curve and ensure tenacity of purpose. Unfortunately groups are subject to many dysfunctions that can be resolved mainly through effective facilitation. A broad variety of skills are required, the most important being based on the ability to effectively communicate.

Storytelling offers an effective means of obtaining feedback from the entrepreneurs in Dwesa as the process is casual and non-intimidating, and is based on a practice that is well established in African

communities.

Prototyping allowed the developer to observe omissions or flaws in the system. The developer was successful in verifying the Xhosa translations of items being sold on the system. He was also able to upload images of the crafts that would be sold on the website, and allocate prices to the items. Unfortunately, the developer's observations were limited as no criticism could be offered by the users. The developer relied solely on his own perception as to how they responded to the system. The process of demonstrating the prototype was not structured in a manner that encouraged criticism. This limited the value of the process as there was little objectivity with regard to the quality of the prototype.

This chapter explores the literature of authors in requirements elicitation techniques and offers various explanations for and solutions to the problems encountered during the activities. Amongst the literature reviewed, the works of George, Hoffer and Valacich (1998) provide guidelines to effectively conduct interviews, Rettig (1994) introduces an argument supporting the application of Low-Fidelity prototyping as opposed to High-Fidelity prototyping, Matts (2003) introduces the concept of negative feedback, and Silver and Wood (1995) discuss the dynamics of working in groups.

The aim of this chapter is to review the literature surrounding requirements elicitation techniques based on the experience applying these techniques in Dwesa. The purpose is to contextualise the techniques to conditions in Dwesa.

3.2 Data Collection Techniques: Interviews, Questionnaires and Surveys

According to George, Hoffer and Valacich (1998), interviews offer higher information richness than questionnaires and surveys, but are more costly to undertake and are more time consuming. Interviews also lack confidentiality, and require a higher level of involvement from the interviewee. Interviews were chosen instead of questionnaires due to the depth of information required during the early stages of the project. Structured group interviews and individual interviews with open ended

questions were conducted with ladies and gentlemen involved in crafts. The aim of the interviews was mainly to discover the nature of crafts and the potential interest by existing crafters in utilising the e-mall to market their goods, as well as to uncover any potential challenges in undertaking the project. The interviews were successful in uncovering the entrepreneurial activities in the area. The interviewees all expressed an interest in participating in the project. However, at the time the interviews were conducted, it was not clear as to whether their interest was genuine or merely expressed out of politeness. Many of the interviewees did however indicate their scepticism towards such initiatives, and added that their hopes had been raised by similar projects in the past and they had been disappointed.

The situation in Dwesa is such that few of the entrepreneurs have attained a tertiary education and many of them do not speak English. Amongst the group that is setting up the E-Commerce platform, all of them have at least an undergraduate qualification, and hardly any of them can speak Xhosa, (The language spoken in the Dwesa area). This complicated the interviewing process as the information gained through interviews went through a translation process where some of the terms or concepts were very difficult to translate. Furthermore, the interviewees seemed to be intimidated by the interviewers and therefore, held back some of their thoughts. There was also poor communication between the interviewer and the translator prior to the interviews. As a result, during the interviews, the author feared that in the effort to encourage the interviewees to participate, the translator used a lot of strong emotive language that may have given the interviewees the wrong perception of the nature of the project.

George, Hoffer and Valacich (1998) recommend that adequate planning of the interview take place prior to the interview and the interviewer should prepare the interviewees for the interview. In this case, the translator should also know what the expected goals and outcomes of the interview are. They add that a checklist, agenda and questions be provided and voice recording equipment should be used if permitted. The interviewer should take notes, which should be reviewed within 48 hours of the interview and a list of unclear points should be presented to the interviewee for clarification.

According to Courage and Baxter (2005), interviews are the preferred elicitation technique when one requires detailed information from individual users. Courage and Baxter (2005) mention that methods such as group task analyses and surveys do not allow sufficient opportunity for a detailed

discussion with the users. George, Hoffer and Valacich (1998) concur with this view and add that surveys permit respondents to remain passive and do not ensure commitment of the respondents. Courage and Baxter (2005) motivate the use of interviews by explaining their role as a foundation for other elicitation tasks. In the case of elicitation techniques such as prototyping and group task analyses, for example, basic domain knowledge derived through interviews may be a prerequisite for the techniques to be successfully applied. In situations where systems are being developed in the absence of formerly established processes and structures, as in the case of rural e-commerce, methods such as observation may not be possible as routine processes upon which to base an understanding of the user's daily tasks may not formally exist or may be difficult to identify. Oppenheim's (1992) description of the role of interviews is consistent with Courage and Baxter's (2005) in that it supports the use of interviews in situations where little is known regarding the subject being examined and greater depth is required.

Courage and Baxter (2005) mention various ways in which interviews can be used to support other requirements elicitation techniques:

1) The use of open ended interviews from which the analyst can define a closed-ended survey
2) The results of interviews can be used to conduct a wants and needs analysis
3) The use of interviews to further analyse the results of elicitation techniques engaged at earlier stages

Oppenheim similarly differentiates the application of interviews at different stages of enquiry by classifying them into two categories; exploratory and standardised. He elaborates that exploratory interviews serve to develop ideas and research theories while standardised interviews aim at gathering facts or statistics. Oppenheim explains that standardised interviews are introduced once the research objectives have been formulated and the wording for the interviews has been perfected after the pilot study has been conducted. Courage and Baxter (2005) add that interviews are inappropriate when information is being sought from large samples of a population. As stated by George, Hoffer and Valacich (1998), as well as Courage and Baxter (2005), interviews are time consuming and expensive. Courage and Baxter (2005) suggest the use of surveys as opposed to interviews, in circumstances where information is required from a large sample, as the results are applicable to the entire user population, rather than just a sample of interviewees. Unfortunately,

according to Courage and Baxter (2005), as well as George, Hoffer and Valacich (1998), this is at the expense of the information richness of interviews. Furthermore, the anonymity of surveys eliminates the ability for the researcher to clarify or follow up on the participant's responses.

3.3 Group Elicitation Techniques: Focus Groups And Joint Application Design

Courage and Baxter (2005) recommend focus groups as a means of drawing multiple viewpoints in a short space of time. They state that focus groups create synergy between participants and create a platform for brainstorming. They also add that focus groups allow researchers to understand the needs of the group as a whole as opposed to the needs of individuals as in interviews. Courage and Baxter (2005) argue that the disadvantage of focus groups is that participants are likely to influence each other's opinions, and the researcher is limited in the number and depth of questions asked. This is also a major shortcoming of Joint Application Design (JAD) and group interviews. The experience conducting group interviews in Dwesa demonstrated the possibility of an unbalanced contribution of stakeholders.

The existence of dominant stakeholders can also be viewed as a strength. The ability for participants to influence one another's views can also contribute to more unified understanding and expectations of the resultant e-commerce platform. It could also promote ownership of the system by the user community as a whole. It was also observed that group sessions have the potential to facilitate the identification of project champions by exposing the participants that have the potential to influence, motivate or organise the rest of the group. The brainstorming that could result from group elicitation techniques may also add depth as it can stimulate debate over the various elements of the system. In the case of the Dwesa E-Commerce platform, where the entrepreneurs will be operating from a common platform, and cooperation amongst entrepreneurs may be necessary, the synergy created amongst participants by focus groups may prove to be of great benefit.

Silver and Wood (1995), mention various ways in which one can benefit from group sessions:

- Groups produce more and better solutions to problems
- Resources are more efficiently utilised by groups

- Inferior ideas are more effectively eliminated through groups
- Productivity is increased by the presence of groups
- Groups have the capability to make better judgements for tasks that can involve random errors
- Groups can achieve more than even the most capable individuals
- Groups have a faster learning curve
- Groups ensure tenacity of purpose by committing members to group decisions
- Groups minimise the sense of responsibility of each individual member and thus tend to make riskier decisions
- Feedback is from multiple sources and is therefore more likely to be accurate

Silver and Wood (1995) also recognise the challenges of group sessions. They summarise them based on the description of dysfunctional group behaviour by Kettelhut (1993) in Silver and Wood (1995), and mention ways of mitigating the challenges.

Table 1: Handling dysfunctional group behaviour. Adapted from Kettelhut (1993) in Silver and Wood (1995)

Behaviour	Description	Mitigation
Groupthink	Groups working together develop group integrity, cohesiveness, identity and energy that can benefit productivity. The group members can close themselves off to unsettling information and make decisions as though they are infallible.	• Restate new information and encourage discussion. • Be cautious of decisions that are made too quickly or ignoring new ideas
Risky Shift	Groups make decisions that pose higher risks than they would do on their own. This allows boundless creativity. Group may make unrealistic decisions that are impossible to implement because no one is individually responsible.	• Evaluate alternatives carefully consider the impact of all the decisions.
Commitment Errors	Group members sometimes make bad decisions and do not change their stance as they do not want to expose the fact that they made a mistake.	• Allow adequate time for the group to estimate the consequences of their decisions.

Goal Setting Errors	Groups often take on tasks that are difficult, and fail to adjust their goals or expectations as their work progresses and may later try to overcompensate for their previous failures by setting even higher goals.	• Challenge all goals and compare them to similar projects.
The Abilene Paradox	The group as a whole adopts a course of action contrary to the desires of most or all of the individual members.	• Encourage silent participants to share their views.

Silver and Wood's (1995) solution to group dysfunction, as described in the above table is largely dependant on good facilitation skills. They recommend that the facilitator should be a sound communicator. This involves possessing the ability to interpret, dissect and integrate a holistic view of what is being said during the session, as well as restate it so that the rest of the group can understand. Furthermore, facilitators are responsible for the preparation of all the necessary documentation that accompanies the process. They caution that the facilitator not confuse their role with that of the project manager. They suggest that the facilitator should not get involved in tasks surrounding program design, coding and testing as they belong to the sphere of application design. Silver and Wood (1995) explain the importance of separating the source of the idea from the idea itself. They further add that the facilitator should be cautious not to be influenced by the manner in which the idea is presented. They elaborate that it is likely that less eloquently phrased ideas may be overlooked despite their relevance. They suggest that the facilitator react impartially to ideas by restating them or writing them on a board where they will appear to be of equal importance. The facilitator should also be able to read the non verbal language of the participants as well as maintain authority and control, while still empowering the group to make decisions.

Silver and Wood (1995) also state that facilitators should be naturally interested and empathetic to what people are saying and should be able to differentiate between a productive debate and a topic

that is diverting from the subject. They add that facilitators should be conscious of hidden agendas, and moments when the participants have reached consensus and decisions should be concluded. The facilitator should furthermore, be able maintain an environment that is organised but can still accommodate spontaneity and uncertainty.

Although the variety of skills needed for effective facilitation cannot all be found in a single individual, Silver and wood (1995) recognise the need to nurture the skills that one does possess, and learn the ones they do not possess.

3.4 Prototyping

A prototype is a working model of a system that exhibits minimal functionality while demonstrating the core elements of the system. According to Silver and Wood (1995), the prototyping process allows analysts to evolve conceptual business requirements into real screen and report designs. This allows users and analyst to see the design early in the lifecycle, and to refine it until it reaches a point where the stakeholders are satisfied. Silver and Woods state that the prototype reflects the business needs more accurately than if the system were to be developed from hardcopy specification or solely through analysts conversations with users.

According to Schach (1998), prototyping results in fewer feedback loops because the likelihood of dissatisfaction during later phases of development is reduced and experience gained from designing the prototype often reveals potential problems that must be avoided by developers in the design phase. Although the initial prototype only exhibits a minimal amount of functionality, the implementation phase is made easier as the prototype provides the design team with some insight as to the products implementation.

The Centre for Technology in Government (1998) provides two broad shortcomings of using prototyping as an elicitation technique;

i. Prototyping can lead to false expectations:

Prototyping often creates the illusion that the system is complete when it is not. The pre-

implementation versions of a system prototype are designed to be one-dimensional structures and the necessary work such as database normalization, documentation, testing, and reviews for efficiency have not been done.

ii. Prototyping can lead to poorly designed systems.

Because the primary goal of prototyping is rapid development, the design of the system is sometimes compromised because the system is built in a series of layers, the prototype being the foundation layer. Issues surrounding the integration of other components may be overlooked and the prototype that had initially been built to be discarded at the end of the requirements phase sometimes evolves into the final product.

3.4.1 High Fidelity and Low Fidelity Prototyping

A prototype of the proposed E-Commerce software was demonstrated to the entrepreneurs in Dwesa. During her observation of the prototyping process, it was noted that too much time and effort had gone into the development of the prototype. This created the possibility of the software developer being resistant to any proposed changes. Schach (1998) mentions how programmers sometimes become attached to their work. The difficulty with this attachment is that the programmers sometimes fail to acknowledge their mistakes in the development of the software. This is one of the difficulties in prototyping, specifically when a lot of work has been put into the development of the prototype. The prototype that was demonstrated in Dwesa took approximately ten months to develop as it was part of a research project. If the prospective users had criticised the prototype, the developer may probably have defended it with little objectivity. This is a scenario identified by certain schools of thought that distinguish between two different approaches to prototyping; high fidelity (Hi-Fi) prototyping and low fidelity (Lo-Fi) prototyping. Instances where a lot of time, effort and resources are engaged to develop the prototype such as with the prototype designed for the Dwesa entrepreneurs are referred to as Hi-Fi prototyping. This often involves the use of high level prototyping tools and the delivery of semi-functional applications that take weeks to develop. Lo-Fi Prototyping on the other hand involves the designing of the prototype on paper, testing it with a selection of users and simulating the way it would be used in reality.

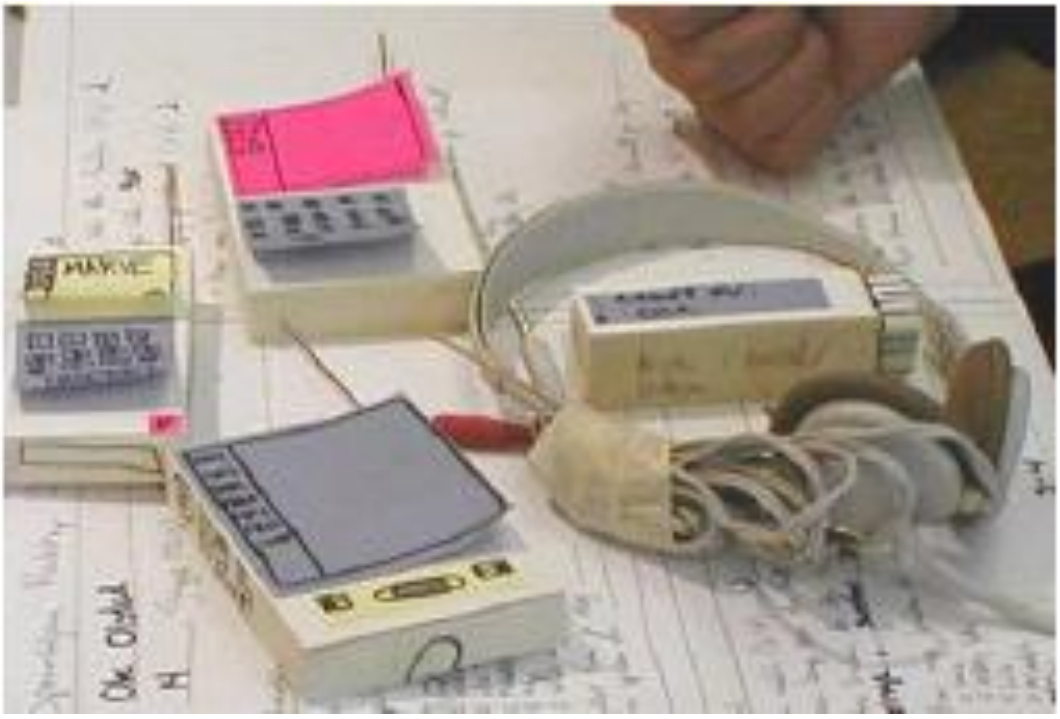


Figure 2: Lo-Fi prototype of wireless peer-to-peer technology (Adopted from Seland and Svanaes)

Figure 3: An example of paper prototyping, Adapted from Rettig (1994)

Rettig (1994) states that irrespective of the high level prototyping tools available, a Hi-Fi prototype may still take too long to build and change, whereas a Lo-Fi prototype can be complete in as little as a few hours. Rettig explains that a single bug encountered in the development of a Hi-Fi prototype may bring its progress to a complete halt. Errors or omissions in paper prototypes on the other hand can be corrected in minutes. Rettig adds that testers of Hi-Fi prototypes often give criticism to issues regarding the aesthetics of the application rather than focusing on the issues surrounding the flow of events, the layout of the controls and the metaphor itself. Rettig also criticises the fact that developers of Hi-Fi prototypes are likely to develop an attachment to their work, and may become resistant to changes. He adds that team members may be conscious of the developer's attachment to their work and therefore less likely to criticise it. For these reasons Rettig advocates the application of Lo-Fi prototyping.

Despite the many strengths of Lo-Fi prototyping, there are also many drawbacks. A study by Lui and Khooshabeh (2003), comparing the use of a Lo-Fi prototype of an interactive automated application prototype found that Hi-Fi prototyping uncovered all the user requirements that Lo-Fi prototyping had uncovered and more. Although raising comments about the user interface is not the only purpose of prototyping, it is an important goal of the exercise. The findings of Lui and Kooshabeh's study revealed that contrary to Rettig's (1994) criticism of Hi-Fi prototyping, the variety of issues addressed while testing the interactive prototype was quite broad, and the aesthetics of the prototype was not overemphasised although the testers did comment on the user interface of the application. However, during the paper prototyping exercise, only one comment was raised about the user interface. This was a major shortcoming of Lo-Fi prototyping. As the study was conducted on a ubiquitous application, Lui and Kooshabeh (1994) concluded that paper prototyping was unable to support prototyping at the level of ubiquitous systems. They argued that the system they had tested was proposed to support natural user interaction through the integration of advanced technologies. The experiments found that paper screens were not of sufficiently high fidelity to convince participants that they could really incorporate the system in their everyday lives.

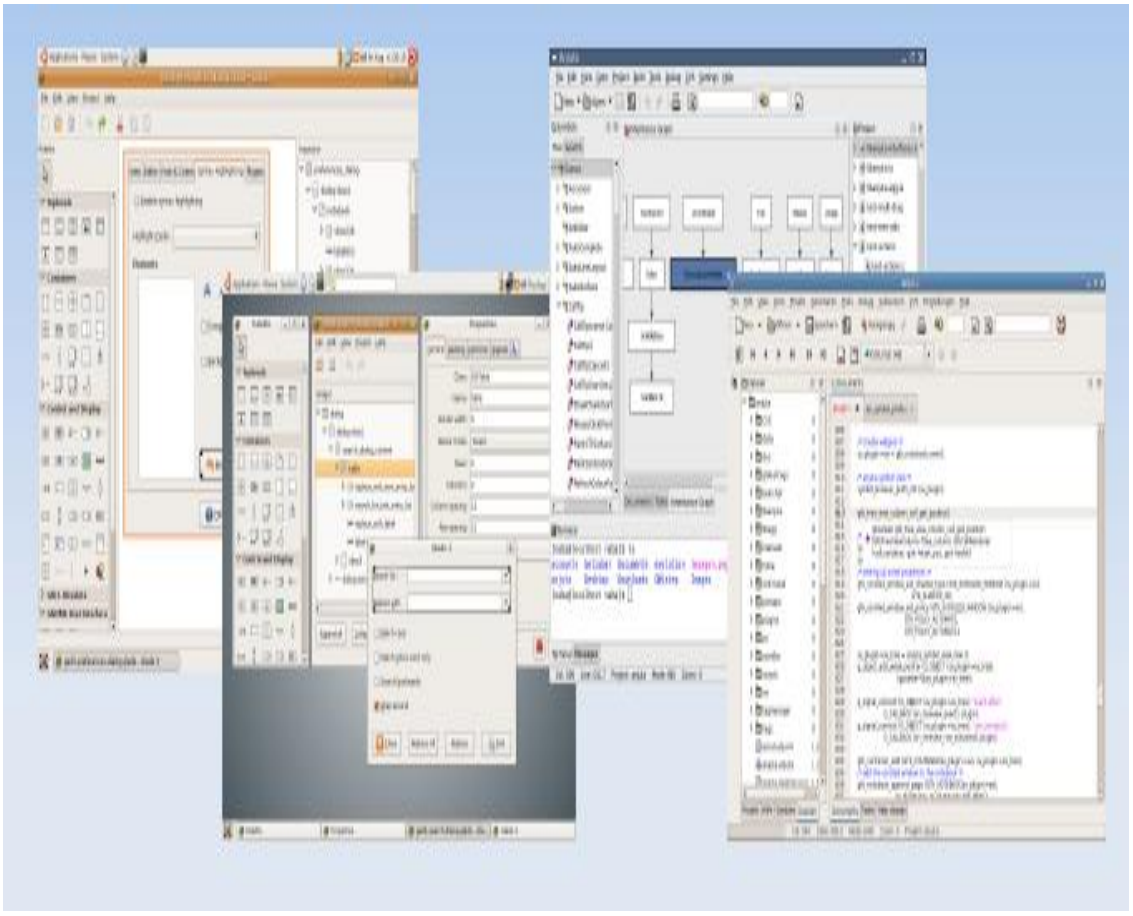


Figure 4: An example of Hi-Fi prototyping, Adapted from Rettig (1994)

The author feels that Rettig (1994) failed to mention the context in which Lo-Fi prototyping could be the preferred technique to Hi-Fi. Lui and Khooshabeh's (2003) study highlighted the unsuitability of Lo-Fi prototyping in the more complex ubiquitous systems. They also revealed that the fact that paper prototypes took less time to implement was not always desirable. In the case of the testing of a prototype of a ubiquitous application, interaction data was required for different times of the day. This made it difficult to collect valid data during typical Lo-Fi prototyping sessions.

This research also notes that both Rettig (1994) and Lui and Khooshabeh's (2003) do not mention the stage of the requirements elicitation process in their studies. Both studies held strong arguments contrasting Hi-Fi and Lo-Fi prototyping, and neither of them were incorrect. These techniques can be used to complement one another at different stages of the elicitation process. Lo-Fi prototyping may be more suitable at an exploratory level when conceptualising the proposed system, while Hi-Fi

can be used to refine requirements at a later stage. Based on the above arguments, although Lo-Fi prototyping is very useful as an elicitation tool, it is likely to be more effective when applied to complement or precede Hi-Fi prototyping.

3.4.2 Negative Feedback

In a situation where the prospective users have had little or no exposure to similar technologies, prototyping offers an effective means of demonstrating the analyst's vision of the system. However, a user that has had no exposure to similar technologies cannot criticise the prototype against any benchmarks or experience. This was the observation made when the first prototype was demonstrated to the users. The technology was completely foreign to its audience and hence no criticism was offered by the prospective users about the prototype. Instead, the user's response was overly positive and instead of reflecting on their impression of the system, it seemed to reflect more on their excitement over seeing their products being uploaded onto the system.

Matts (2003) argues that traditional waterfall approaches to software development are inclined towards gaining positive feedback from their audiences. He explains the following reasons for this emphasis on positive feedback:

- **Pressure to stay on schedule**: Negative feedback may delay signing off any progress in the software development cycle. As the software being tested in Dwesa is part of a Masters Research project, the developers have to adhere to strict deadlines. This is likely to affect the level of sensitivity or objectivity with regard to the user's response or to errors and omissions in the prototype.
- **Insufficient information or ability to articulate negative feedback**: The users might be doubtful about aspects of the system but are unable to articulate their dissatisfaction. This is often the case with missing requirements as they are more difficult to identify than incorrect requirements. In the case of the prototype demonstration in Dwesa, the users had a limited background with the technology and therefore were unable criticise it.
- **Signing off a project relieves project members of pressure and responsibility**: Being responsible for holding up the project is perceived in a worse light than being responsible for

signing off on incorrect requirements. Likewise, the priority of the researchers on the project in Dwesa was to complete their Masters dissertation on time rather than to ensure that the end product could serve the user's needs sustainably.

- **Fear of the opinion of others:** Users do not want to appear less knowledgeable by asking obvious questions, nor do they want to upset the analyst by criticising their work.
- **Lack of understanding:** It is easier to give positive feedback than it is to give negative feedback as negative feedback requires a proper understanding of the material.

The author attributes the abovementioned reasons to the failures in the prototyping process she observed in Dwesa. Insufficient information or ability to articulate negative feedback, fear of the opinion of others and lack of understanding certainly contributed to the failure of the prospective users to criticise the prototype. On the other hand the pressure to stay on schedule and need to sign off the project may have influenced the fact that the developer did not sufficiently probe for more feedback. Matts (2003) argues that keeping on schedule and signing off the project on time is often at the cost of gathering correct requirements. For this reason, he advocates the prompting of negative feedback to allow objective and constructive criticism of the software. He recommends the following methods as a means of gaining this feedback:

Face-to-Face Communication

Because a vast component of our communication is non verbal, being physically present to observe factors such as intonation and body language can contribute significantly to the understanding of the user's opinion of the software. Matts (2003) suggests that one must take note of hesitations, shrugs, sidewise glances or any other movements that indicate negativity towards the system. This could be particularly useful in a rural ICT project such as the one being conducted in Dwesa where the verbal feedback is limited by language and background. The body language of the users can reveal more information to the analyst. However, the observation of non verbal communication can yield subjective opinions and should thus be practiced with caution.

Fuzzy versus Hard

This refers to the manner in which material is presented. Fuzzy material is more vague and incomplete while hard material is more refined. Matts (2003) explains that fuzzy material is more

likely to yield feedback than hard material. He elaborates by saying that users often feel discouraged to criticise complete work. Matts (2003) classifies paper prototyping in the category of fuzzy material. The incomplete nature of paper prototypes makes them useful in encouraging feedback.

Storytelling

Stories help to establish context, and guide the listener to ask questions concerning the way the solution was arrived at instead of focusing merely on the solution. According to Matts (2003), storytelling allows the reviewer to contribute to the narrative and ask questions about why certain decisions were made. This offers the listener a more complete understanding of the solution.

Celebrate Failure (of the Model) / Break the Model

This involves a shift in the attitude of analysts, where instead of defending or seeking to validate models that they have developed, they should praise efforts by users to criticise their work, and celebrate the exposure of flaws in the models.

Be sensitive to "Bad Smells"

This requires that analysts take note of all warning signs of anything that could have a negative bearing on the project such as;

- Business users hesitating before answering positively.
- The existence of many-to-many relationships in business models.
- Unclear relationships among business objects.
- The business users fail to understand even the straightforward example of the model.
- The business users disagree on how the system should work.
- The business users are constantly changing their minds.
- Inability of business users to express what they want.

State what the system will not do

When stating the scope of a system, Matts (2003) recommends that one explicitly states the functionality that the system will have as well as what it will not have. He explains that this forces the analyst to search for any possible options that may be missing.

3.5 Stories And Narratives

Stories, as defined in Alvarez and Urla (2002) are narratives about specific past events. They are embedded and fragmented processes within which gaps are filled by narrators and their audience. Alvarez and Urla (2002) differentiate between stories and narratives by classifying stories as a form of narrative where events follow a chronological and linear sequence. Other genres of narratives exist apart from stories. These include habitual narratives (repetitive events with no climax) and hypothetical narratives (events that did not happen).

Erickson (1996) explains that effective stories prompt gestures and related narrations from their audience. He explains that people who are reluctant to contribute to a topic are more likely to participate by telling stories because they are not claiming to be knowledgeable on the matter, nor can they be criticised of dispensing false information as they are merely narrating a story. Storytelling offers an effective means of obtaining feedback from the entrepreneurs in Dwesa as the process is casual and non-intimidating, and does not require them to commit themselves to their comments as the characters they discuss can be hypothetical.

Stories can be conducted as a precursor to more formal approaches such as observation, structured interviews, and task analyses as they provide the analyst with an understanding of the domain within which the user operates, from the users own perspective, and provides a platform for further exploration (Erickson, 1996). Erickson (1996) also adds that although stories are very useful during the early stages of design, they can also be used as mechanisms for communicating with the organisation once the exploratory stage of design is complete. He explains that many stakeholders at this stage will not have the time or interest to examine design documents. Stories are very effective in summarising and communicating this kind of information.

According to Erickson (1996), during an interview, listeners provide verbal and non-verbal messages prompting the interviewee to continue, repeat, elaborate or omit information. The Interviewer responds to these prompts and the audience immediately express how they have assessed the speaker's meaning. Ultimately the interaction shapes the narration and interpretation of

the story. He adds that the comments and responses that emerge from stories act as correcting mechanisms, while continually revealing the development of shared meaning. The story is thus dynamically produced, challenged, and revised during its narration.

According to Erickson (1996), the process of collecting and narrating stories can help users feel more comfortable as participants in the design process as they make formal processes such as interviews seem less threatening. Stories according to Erickson (1996), also promote collaborative work and understanding within the design team. One of the strengths of storytelling mentioned by Erickson (1996) that is particularly relevant to Dwesa, is that compared too many group techniques, they do not require much experience or training to execute. He states that team members from any background can be made part of the storytelling process and adds that the storytelling process sensitises participants to the usage domain, helps to identify questions and issues to explore in discussions with users, and provides examples of real experiences that can be used to conclude debates about vaguely defined abstraction.

The casual, non-intimidating nature of storytelling makes it a potentially powerful tool when conducting research in Dwesa. Storytelling is already a well established means of communicating knowledge within many African Communities. This could benefit the analyst by allowing them to communicate with the community members in a format that is not foreign to them.

3.6 Conclusion

Interviews proved to be an effective means of understanding of the needs of the community. Adequate preparation prior to conducting the interviews is necessary. This includes preparing the translator to ensure that the aims of the interview are appropriately addressed. Different types of interviews exist for different stages of enquiry. Exploratory interviews are detailed and personal and are useful during the early stages of research to develop ideas, while standardised interviews target a broad range of respondents and are useful once the research problems have been established. In some cases the response may be challenged by the fact that respondents feel afraid to voice their opinions to the interviewers. Storytelling provides a relaxed and non-intimidating format of enquiry. It is presented in a manner that is familiar to many African cultures and is thus worth exploring in

Dwesa. Group techniques are a powerful way to empower user to make decisions. The success of group techniques is largely dependant on the skills of the facilitator.

Prototyping allowed the developer to translate English terms into Xhosa, exhibit the manner in which the system would operate, upload images of the crafts, and allocate prices. The prototype demonstrated in Dwesa was an example of High-Fidelity (Hi-Fi) prototyping. It took a long time to develop, and was fully interactive. One of the drawbacks of this was that it created the likelihood of an attachment to the design by the developer, and hence reduced their objectivity with regard to negative feedback. In this case no negative feedback was offered by the users due to their limited knowledge and ability to criticise it. Actively encouraging negative feedback is one of the measures that may assist the developer to get an objective response regarding the prototype. Low-Fidelity (Lo-Fi) prototyping is a technique whereby the prospective users interact with illustrations on paper of the various screens of the system. It takes less time to develop than a Hi-Fi prototype, and can be modified easily during interaction with users. Low-Fidelity prototyping is useful when demonstrating less complex systems. In more complex systems, analysts can apply both Hi-Fi and Lo-Fi prototyping, in such cases Lo-Fi prototyping would be beneficial during the early conceptual phase, while Hi-Fi may be used to refine requirements at a later stage.

All the techniques reviewed can be applied effectively within a rural context, however, the analysts need to take into consideration the objectives of the stages of software development within which the techniques are being applied, as well as the variables surrounding the users that could negatively influence the elicitation activities. Analysts also need to ensure that they have the necessary skills and resources to engage in the specific requirements elicitation techniques. These include competent facilitators, interpreters and recording devices. Although the review of elicitation techniques can inform the process of determining which techniques to apply, the fact that no single technique reviewed can provide an infallible solution to the challenges of requirements elicitation in a rural setting necessitates further exploration into methodologies and approaches that can provide better guidance for the process of requirements elicitation. The chapter that follows will review a selection of requirements elicitation methodologies with respect to their ability to sufficiently guide requirements elicitation in a rural context.

Chapter 4

A Review of Software Development and Requirements Elicitation Approaches

4.1 Introduction

When defining an approach to requirements elicitation that can be implemented in the context of software development for rural users, it is ineffective to evaluate the techniques unilaterally. Requirements elicitation cannot be decoupled from software engineering, nor is it restricted to one stage of the software development lifecycle. This chapter explores how software development practices have evolved over the past decades, and describes the level of integration between requirements management and the software development lifecycle. Four requirements elicitation approaches are compared and contrasted against one another in this chapter. The aim is to provide an understanding of the goals of the respective software development models in order to establish a foundation upon which a new model for requirements elicitation can be defined.

4.2 The Evolution of Software Development Approaches

Systems development methodologies have evolved over the years in response to the changing needs of software users, and the associated shift in demands on software developers. This section provides a review of software development methodologies with the goal of developing a perspective on the interaction between the process of requirements elicitation and the software development process.

4.2.1 Addressing Complexity in Systems Development

The need to address the complexity of systems development through defining systems development methodologies was formally recognised as early as 1956. In the 1970s, the traditional approach to software development evolved into a more structured, efficient and disciplined process. Mcnurlin and Sprague (1993) described this transformation as a redefinition of systems analysis and programming from a free form art, to a craft. Unfortunately the negative consequences were the long development times, lack of user involvement, and inflexible resulting systems.

4.2.2 The Waterfall Model

The Waterfall Model was first developed by Royce (1970). The Waterfall Model is composed of four development stages; Requirements Analysis, Specification, Design and Implementation. These phases are followed by the operations mode and subsequently the retirement mode. During the requirements phase, the system requirements are determined and verified by the client and members of the Structured Quality Assurors (SQA) group. This marks the end of the requirements phase and the beginning of the Specifications phase. During the Specifications phase the requirements are compiled into a requirements specification document. The specifications are assessed by the SQA group and once approved, delivered to the client for verification. The specifications document serves as a binding agreement between the client and the developers regarding the specifications of the system. The design phase follows the specifications phase and is based on the output from the specifications phase. During this phase, omissions, contradictions, or ambiguities in the specifications may surface. The Waterfall Model contains feedback loops at the end of each development phase. This allows for revision of system requirements, specifications and design. In the implementation phase, testing allows any flaws in the design document to surface, and the design is revised through the feedback loop. All modules are implemented and documented in the implementation phase, and then integrated to form a complete product in the integration phase. Modifications to preceding phases can be initiated and documented during the integration phase.

The Waterfall Model is still a commonly used approach to software development. It revolutionised software development by introducing and enforcing discipline to the process. Documentation exists at the end of each phase as a direct consequence to the manner in which discipline is enforced, Schach (1998). The Waterfall Model emphasises verification or testing as a part of each phase, as opposed to testing the end product. This minimises maintenance problems. Furthermore, the lifecycle encourages that stakeholders sign off at the end of each phase to minimise legal conflicts.
The document driven nature of the Waterfall Model can also be a shortcoming. The audience of the specification documents are often inexperienced in reading software specifications as they are often written in complex IT specific language that is not fully understood by the client. According to Schach (1998), amongst the shortcomings of the Waterfall Model is the fact that its document driven nature means that the client only sees a working product at the end of the lifecycle. He suggests that this may lead to an end product that does not meet the client's expectations. Furthermore,

according to Boehm (1988), the overemphasis on documentation may compromise usability especially in the development of interactive systems. Boehm (1988) also adds that less documentation is required in the advent of Fourth Generation Languages (4GLs). Schach (1998) argues that the Waterfall Model may not be a suitable model to follow in instances where products need to be developed over a short space of time. Some of the shortcomings of the waterfall approach are addressed in the Spiral Model and the Rapid Prototyping Model.

4.2.3 The Rapid Prototyping Model

Data driven development and group design sessions emphasised the need for early detection of potential problems in the systems being developed and more rapid systems development. By the 1980s this was made possible through the use of 4GLs and prototyping. At this point analysts were also beginning to recognise the importance of usability.

The main difference between the Waterfall Model and the Rapid Prototyping Model is that prototyping is the central means of eliciting the requirements. During the first phase of the lifecycle, a prototype is developed and presented to the client. The client and potential users are then given the opportunity to experiment and interact with the prototype, and offer criticism or recommendations regarding it. When all the stakeholders are satisfied with the features exhibited by the prototype, the system can be documented into a specification document. The Rapid Prototyping Model is identical to the Waterfall Model from this point onwards

The advantages of prototyping were identified as early as the 1980s. Burns and Dennis (1985) stated that prototyping results in a higher level of user satisfaction due to the high level of involvement of the user in early stages of development and the gradual refinement of the system based on the user's responses to the prototype. Other advantages include the fact that there are fewer feedback loops because the likelihood of clients being dissatisfied during later phases of development is reduced. Experience gained from designing the prototype often reveals potential problems that must be avoided by developers in the design phase. Although the initial prototype only exhibits a minimal amount of functionality, it still provides the design team with some insight as to the products implementation (Schach, 1998).

Burns and Dennis (1985) argue that prototyping results in less efficient systems than the traditional approach due to the insufficient analysis and design in early stages of development. This view is supported by The Centre for Technology in Government (1998), who state that because the primary goal of prototyping is rapid development, the design of the system can sometimes suffer because the system is built in a series of layers without a consideration of the integration of all other components. Initial software prototypes are built to be discarded eventually, and thus, attempting to produce a solid system design based on a prototype may prove to be problematic.

The centre of technology in government (1998), also state that prototypes can often be misleading as the client gets the impression that the product is complete, yet it is not. This is because, when using the prototyping model, the pre-implementation versions of a system are simply one-dimensional structures. The necessary technical foundations such as database normalization, documentation, testing, and reviews for efficiency have not yet been done.

4.2.4 The Spiral Model

The shortcomings of the Rapid Prototyping Model prompted analysts to explore mixed methodologies in systems design. The Spiral Model emerged as a result of this. The Spiral Model accompanies the stages in the Waterfall Model with a cyclical review of specific factors. These include system objectives, constraints, alternatives, risks, risk resolution, risk resolution results, planning for the next phase, and commitments. The risk analysis orientation of the Spiral Model is one of its major strengths. This feature makes it highly adaptive and useful when being applied in unconventional contexts as it forces stakeholders to clarify the objectives of each phase and instils the discipline of assessing the potential constraints and mitigations of each phase before attempting to engage in the activities of that phase. The main advantage that the Spiral Model has over the Waterfall Model is that it is not driven by large quantities of documentation. Instead the cycle is supported by risk assessments at the start of each phase that contribute to the planning of the phase, and a prototype during each cycle to demonstrate the analyst's vision of the required system. It makes the cycle adaptive in the sense that it allows analysts to respond to the challenges that are relevant to the particular system being developed. This view is expressed by Boehm (1988), who explains that the Spiral Model accommodates most models as special cases, and determines which combination of previous models best suits a specific software development situation. The Spiral

Model is also quite specific in the fact that prototyping is used at each phase as a means of specification. This minimises the problem of confusing specifications. Unfortunately the Spiral Model elaborates the main shortcoming of prototyping as the prototype has the ability of evolving into the final product without being sufficiently documented. The prototyping feature of the Spiral Model does however help in addressing usability issues in systems design as validation of each phase is more practical.

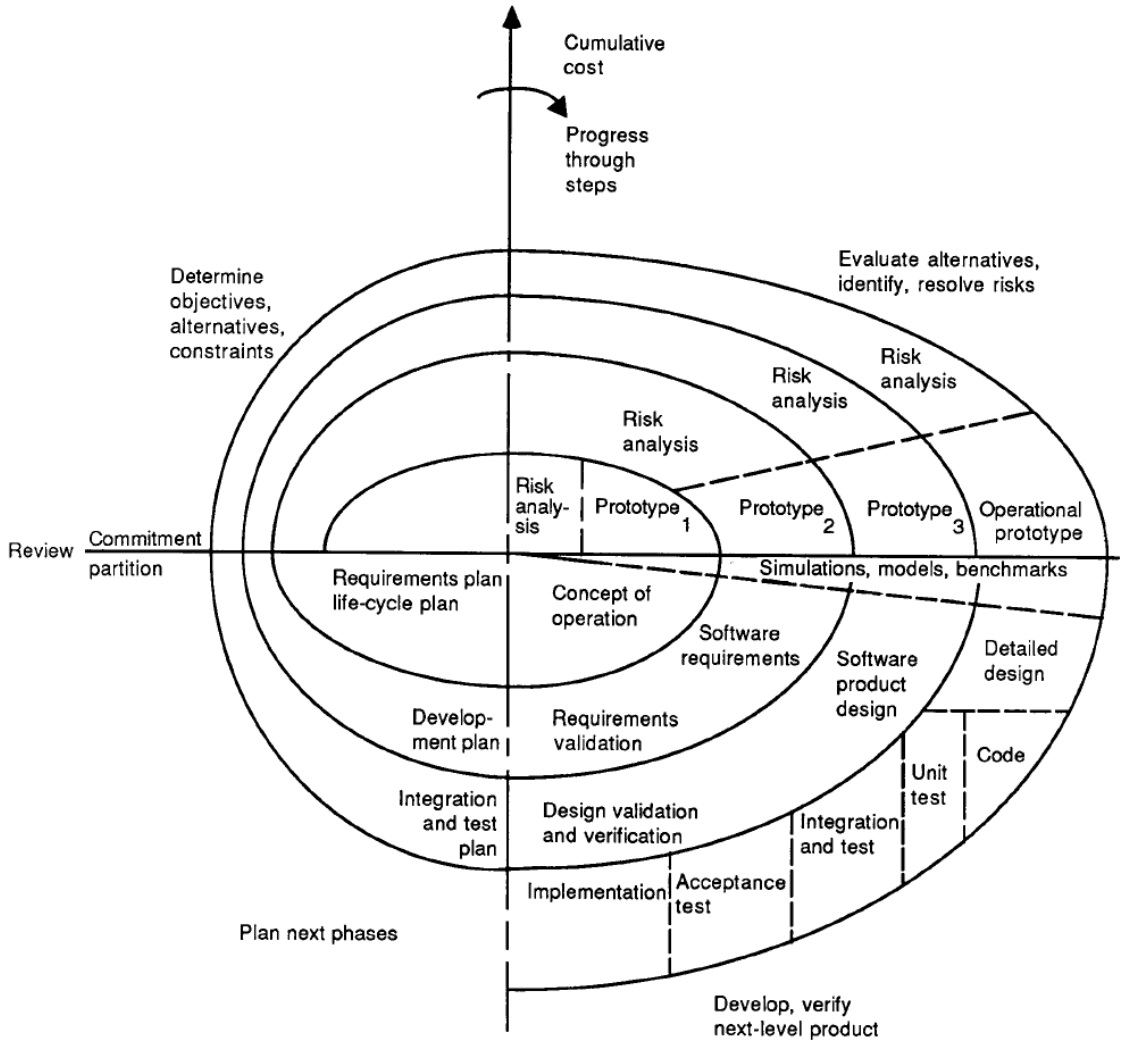


Figure 5: An example of the components of a single stage of the Spiral Model (Boehm, 1988)

4.2.5 Computer Aided Software Engineering, Object Oriented Software Engineering and Fourth Generation Languages

Towards the mid 1980s, Computer Aided Software Engineering (CASE) and Object Oriented Software Engineering (OOSE) emerged and further transformed software engineering practices. CASE allowed the automation of existing structured development techniques, and OOSE provided efficient means of developing systems, and encouraged concepts such as reuse of code and localisation of the effects of system changes. According to Mayhew (1999) OOSE is characterised by its highly iterative approach to modelling each phase of development. Mayhew adds that instead of exhaustively defining functional requirements before design begins, and specifying design exhaustively before construction begins, OOSE is made up of small subsets of functionality that are followed through the various models, validated, and then modified. OOSE and CASE tools can be applied to any of the methodologies or lifecycles defined in this chapter. Fourth Generation and Languages, which are programming languages closer to human languages, simplified the software development process even further and made way for approaches such as agile software development and extreme programming.

4.3 Targeting The User In Systems Design

4.3.1 User Centred Design and Participatory Design

The UCD approach combines various methods to assist engineers and developers in addressing the issue of usability in systems design (Rauterberg, 2003). Involving the users of a new system in the development process, according to Myers (1997), allows the developer to anticipate or identify factors that may contribute to resistance. The application of a UCD approach to software development is likely to direct developers to the questions that must be answered if products are to be effectively adopted and utilised in their intended market (Rauterberg, 2003).

Participatory Design is an approach that was established in Northern Europe. It is described in George, Hoffer and Valacich (1998), as an alternative to existing Software development methodologies. In Participatory Design the main focus is the improvement of the work lives of users. This is achieved, according to George, Hoffer and Valacich (1998), through the continuous involvement of users in the development process. Rotmans (1998), in George, Hoffer and Valacich

(1998), describes Participatory Design as "an umbrella term describing approaches for assessments in which non-scientists, such as policy people, stakeholders, even lay people play a role". In Participatory Design, the analysts are considered to work for the users and the organisations management and consultants are there to offer advice rather than control.

The evolution of software development practices over the past five decades has seen a shift in focus from minimising the complexity of the software development process, to enhancing the user experience and optimising the value yielded from a software system. Although classical software engineering models have resulted in an improved software development process, they have failed to explicitly address the needs of the user in their structures. The experience gained through eliciting requirements in Dwesa proved that evaluating requirements elicitation techniques provides a theoretical understanding of their application, but in order to effectively improve the requirements elicitation process, a methodology that can be adapted to the context of its application is necessary. Few frameworks exist that appropriately support the process of determining requirements in general and less so in the case of rural software development.

The ultimate goal of software development is to create software systems that can solve the users' problems, leverage their opportunities, or satisfy their needs (Davies and Hickey, 2004). While software development may follow a broad range of different methodologies or frameworks, requirements activities tend to be performed at the beginning of the software development (Davies and Hickey, 2004). Davies and Hickey argue that as the needs of the users or the organisation changes, so too must the requirements. This must be accommodated in the software development methodologies.

4.4 The Requirements Phase of Systems Development: Different Approaches To Requirements Elicitation

Today the requirements phase is broadly recognised as one of the most critical and challenging phases of systems development. Brooks (1987); Kaarilahti, Pekkola and Pohjola (2006) and Wiegers (1990), amongst others, attribute failures in many software development projects to the inadequate planning and execution of the requirements phase. Leffingwell (1997) states that

between 40% and 60% of errors in software projects can be attributed to errors in the requirements phase. He adds that ambiguous user requirements can result in rework that could amount to as much as 40% of the total software development costs, and further states that between 70 and 85% of revisions to software development projects result from requirements errors. A study by Boehm, (1981) found that it costs 68 times more to make corrections to a software product after it had been in operation, than if it were to be corrected during the requirements phase. According to Davis (1995), the five most frequently reported causes of software cost estimation relate to requirements engineering. Although these figures are based on isolated studies, and do not necessary reflect a universal truth, they have all been conducted by highly recognised authors in the field of requirements management, and they do highlight the importance of the requirements phase. Unfortunately, software development teams are almost always restricted in terms of time, and hence often focus more on the technical aspects of software development at the expense of the requirements phase. Wiegers (1996) suggests that emphasis be placed on early stage quality practices because a good quality requirements process has a high leveraging effect on the outcome of the software development process.

Literature surrounding requirements elicitation has been largely centred on guidelines by experts or academics for applying or improving specific requirements elicitation techniques. However, there are very few models that illustrate the process of requirements elicitation in general (Browne and Rogich, 2001). This section evaluates four requirements elicitation models with the intention of understanding the various perspectives of requirements elicitation, and developing a requirements elicitation methodology that can be applied in a rural context.

4.4.1 Browne and Rogich's Task Characteristics Model

Browne and Rogich (2001) provide an approach that generally models the elements of requirements elicitation. They refer to this model as the "Task Characteristics Model", and argue that it provides the analyst with a framework for determining what the requirements are, and hence enables them to understand what their goals should be. They argue that the strength of their model is that it perceives the requirements as more than mere documentation of data elements in a business process. Instead they argue that their model recognises the role requirements elicitation plays in the identification of goals, assumptions, opinions, and desires of users.

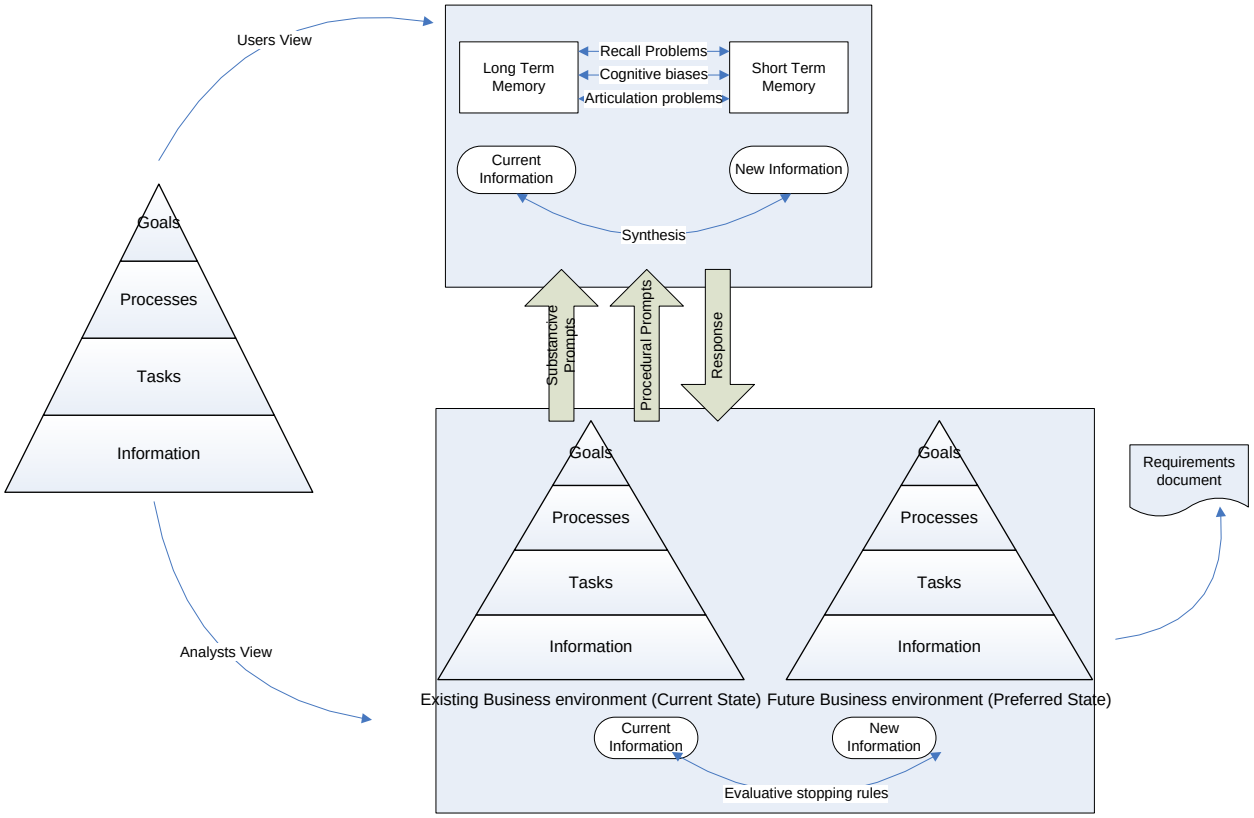


Figure 6: Browne and Rogich's (2001) Task Characteristics Model

4.4.1.1 Elements of the Task Characteristics Model

i. Stimulus

According to Browne and Rogich (2001) the need for systems development is typically stimulated by the existence of an organisational problem or opportunity. Browne and Rogich (2001) categorise the problems or opportunities into four classes; organisational goals, business processes, tasks that must be performed within the processes to achieve the goals, and information necessary to inform tasks. They represent these classes in a pyramid structure in the model for two reasons. Firstly, the hierarchical representation places general goals at the top and specific information at the bottom. Secondly, they recognise that in any systems development initiative, there will be fewer goals than processes, fewer processes than tasks, and fewer tasks than information. Thus the pyramid structure is a useful means for organising and describing types of requirements.

According to Browne and Rogich (2001), the user's problem space and the analyst's problem space can each be conceptualised as a hierarchy of goals, processes, tasks and information represents the problem space.

ii. User's Problem Space

According to Brown and Rogich (2001), during the evocation of requirements, the user's problem space contains information recalled from long-term memory in response to needs that exist in working memory. They state that the process of evoking requirements is hampered by factors such as failure to recall information from long-term memory; cognitive biases and inabilities to articulate routinised procedures that have become automated. They add that the users are faced with both the challenge of recalling details of the current environment, as well as the challenge of envisioning the future preferred state of the environment.

iii. Analyst's Problem space

According to Browne and Rogich's (2001) model, the successful gathering of requirements is dependant on both the elicitation of a conceptualisation of the problem, as well as a visualisation of a preferred future organisational environment from the user. This process requires the analyst to organise the user's responses in order to conceptualise current and future states of the organisation. As the analyst begins to build an understanding of the user's needs, partial solutions to the problem are captured. Browne and Rogich (2001) explain that these partial solutions become subject to extensive mental simulations and reformulations that are eventually terminated by the analyst's use of evaluative stopping rules. At this point the analyst feels he has obtained enough information and thus captures the information in a requirements document.

iv. Interaction between User's Problem Space and Analyst's Problem Space

Analysts make use of various types of prompts to stimulate the user's thought processes in order to improve their understanding of the problem and to identify missing requirements. Browne and Rogich's (2001) model is founded on the evaluation of two context independent prompting techniques, procedural prompts and substantive prompts. Both types of prompts serve the intention of aiding users and analysts to overcome cognitive challenges associated with the requirements

elicitation task.

v. Procedural prompts

Procedural prompts are designed to elicit information that might otherwise not be evoked due to cognitive challenges. Browne and Rogich (2001) use the example of the question "What kinds of things can people do now that they might not be able to do when using the system?", this question is aimed at prompting users to re-examine their understanding of the system and to scrutinize it critically. Browne and Rogich (2001) argue that such prompts have been shown to reduce judgmental biases. Asking users to summarize their opinions, another example provided by Browne and Rogich (2001), can help overcome the limitations of short-term memory. Procedural prompt can also ask a user to create a scenario describing how he performs a task. Scenarios according to Browne and Rogich (2001) have been shown to help people articulate information that they would otherwise struggle to evoke.

vi. Substantive prompts

Substantive prompts are aimed at eliciting specific types of requirements. Substantive prompts should stimulate users' memory structures in a manner that will improve their ability to recall knowledge relevant to system requirements.

Table 2: Examples of procedural and substantive prompts, adapted from Browne and Rogich (2001)

Question	Type of Prompt
What would your customers want the system to do?	Substantive Prompt
Why would your customers not want to use the system?	Procedural Prompt (Causal counterargument)
What can be done to overcome these negatives?	Procedural Prompt (Causal counterargument)
What would your employees want the system to do?	Substantive Prompt
Summarize everything you want the system to do?	Procedural Prompt (Summarization / feedback)
What must the customers do to use the system?	Substantive Prompt
Can you think of a situation where the customer would have difficulties using the system?	Procedural Prompt (Scenario building)
What can be done to overcome these problems	Procedural Prompt (Causal counterargument)
Summarize the steps for using the system	Procedural Prompt (Summarization / feedback)

vii. The Requirements Document

The requirements document, which includes information elicited from users and gathered from other sources, represents a description of a future state of the organisation that, when implemented, includes a system that is aimed at enabling the organisation to achieve the goals identified.

4.4.1.2 Strengths of Browne and Rogich's Task Characteristics Model model

Browne and Rogich (2001) argue that their Task Characteristics Model was empirically evaluated in a comparison between two other well established questioning techniques (interrogatories and semantic methods). They state that the study revealed that the task characteristics prompts elicited significantly more requirements than both the semantic prompts and the syntactic prompts.

Browne and Rogich (2001) argue the following strengths of their model:

1. It offers analysts a clear description of the elicitation task, and aids practitioners in understanding both the problems and opportunities typically found in the relationship between user and analyst.
2. It provides analysts with a prompting technique for eliciting requirements that is based on theoretical knowledge from several domains.
3. It is empirically tested and has been found to be effective in eliciting more requirements than other prompting techniques.
4. It has resulted in an enhanced taxonomy of generic requirement categories for organizing information gathered in the analysis stage of system development that can guide practitioners in the identification and capture of requirements.

4.4.1.3 Weaknesses of Browne and Rogich's Task characteristics model

Browne and Rogich's study was conducted within a single organisation, with well established business processes. Furthermore, one of the criteria for selection of the participants of the study was that they must have used a computer system as part of their job function and have had experience using databases. This suggests that the model's success has only been proven under very specific conditions. Furthermore, the conditions under which the model's success is based are vastly different to typical rural conditions. The study also tested the prompting techniques as opposed to

the model, and the prompting techniques were only evaluated against other prompting techniques as opposed to being tested against a range of requirements elicitation methods. The study proved that the model is effective in eliciting more requirements, but failed to prove that the quality of the requirements elicited was superior. This research is of the view that the quantity of requirements obtained does not necessarily imply the success of the requirements elicitation task.

These criticisms of the Task Characteristics Model are only intended to point out the fact that the model has been proven to be effective under very specific conditions. Unfortunately, when considering the model in a rural scenario, where the business processes have not yet been formerly established, the system is a completely foreign concept and the users have no prior experience with similar systems, such a model will be very difficult to implement. The prescription of specific prompting techniques adds an undesirable rigidity to the model in situations where limited knowledge of the various problem spaces exists.

4.4.2 Davies and Hickey's Requirements Elicitation Technique Selection model (2002:2004)

Davies and Hickey (2002:2004) suggest that the requirements elicitation process should involve iterative steps to select an appropriate technique given the profile of the stakeholders, problem domain, and project details. They define their approach as "knowledge intensive" as it is founded on the principle that ample knowledge of the environment in which the technology is being implemented is required in order to effectively elicit the user requirements. This knowledge refers to knowledge of the users and stakeholders, the problem and solution, and the project domain.

Davies and Hickey (2002:2004), argue that most requirements elicitation fall under either of two broad classifications; those that capture a specific methodology or technique, and those that model elicitation in general. Under each of these classifications, Davies and Hickey have identified various shortcomings.

i. Elicitation approaches that capture a specific technique or methodology

Davies and Hickey criticise the manner in which requirements elicitation approaches that are

founded on the use of specific methodologies prescribe predefined techniques for requirements elicitation. They argue that these models do not specify the factors used to determine the techniques to apply to determine the user requirements, and hence imply that a generic solution can be effective in almost any scenario.

ii. Elicitation Techniques that Model the Elicitation Process in General

The second classification described by Davis and Hickey (2004) are often founded on unspoken assumptions, and do not discuss the role of knowledge in performing elicitation or in selecting elicitation techniques. The knowledge referred to by Davis and Hickey includes firstly the current problem, solution, and project characteristics, secondly, the awareness of which requirements are known and which are still to be determined, and finally, knowledge of the relationship of the current problem, solution, and project characteristics and the state of the requirements to the selection of an elicitation technique.

Davis and Hickey (2002) recommend a model that addresses the abovementioned shortcomings.

4.4.2.1 Strengths of Davies and Hickey's Model

Davis and Hickey argue that the strength of their model is that it explicitly highlights the role knowledge plays in performing both elicitation and elicitation technique selection. They add that the model also provides a unified framework for understanding the purpose and role of requirements elicitation in software development and shows what assumptions existing elicitation methodologies make about the situation. The model makes it easier to tailor existing methodologies for unique situations, and allows analysts to create new elicitation methodologies, by defining situational characteristics and then observing and recording the resultant instances of methodologies.

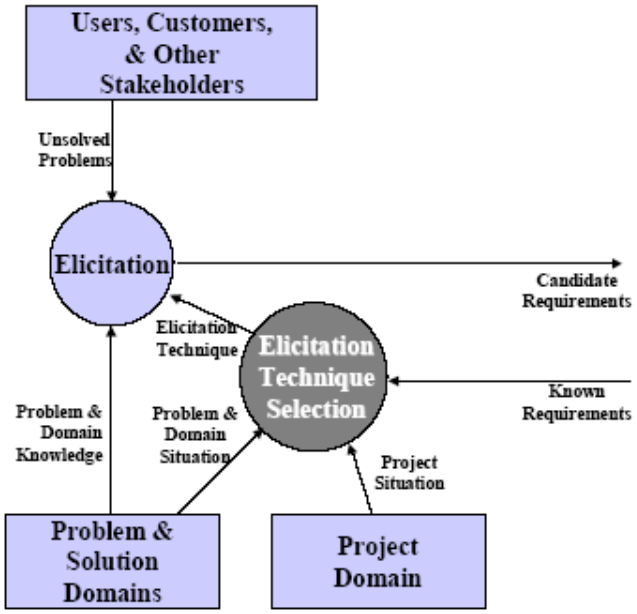


Figure 7: A model for knowledge intensive software development processes. Davies and Hickey (2002:2004)

4.4.2.2 Weaknesses of Davies and Hickey's Model (2002:2004)

Davies and Hickey (2002:2004) base their model on the fact that the choice of requirements elicitation techniques is highly dependent on the knowledge of the environment within which the software is to be applied. They explicitly state that this knowledge includes knowledge of the problem and solution domain, and the project characteristics. They also state that knowledge of the users and other stakeholders contributes to the elicitation process, but do not make any reference to how knowledge of the users can affect the elicitation technique selection process. This, in the author's view is the first shortcoming of the model.

It is possible that in studies conducted by Davies and Hickey, the profile of the users and stakeholders had no bearing on the technique selection process. However, taking rural ICT users into consideration, it may be necessary to include the user and stakeholder profile as one of the factors that determine the choice of elicitation techniques. The previous chapter explored various requirements elicitation techniques and discusses their application in a rural context. The main characteristic that differentiates rural ICT implementations from urban ICT implementations is the users. Rural ICT use is often challenged by issues such as low literacy levels, little or no prior exposure to technology and the inability to effectively communicate in any language other than the mother tongue. These factors will almost certainly affect the choice of elicitation techniques.

Another shortcoming of Davies and Hickey's (2002:2004) model is that it does not take into consideration the software development stage in which the techniques will be applied. This is also likely to have a bearing on the effectiveness of certain techniques. One cannot identify the factors that affect the technique selection process exhaustively, as many of these factors may be context specific.

4.4.3 Christel and Kang's Requirements Elicitation Process Model (1992)

Christel and Kang (1992) categorise the problems associated with requirements elicitation as problems of scope, understanding and volatility. They attribute the existence of ambiguous, incomplete, unverifiable, inconsistent, irrelevant, incorrect requirements that do not reflect the stakeholders' needs and objectives to these problems.

i. Problems of Scope

According to Christel and Kang (1992) elicitation techniques need to be broad enough to define the boundaries of the proposed system, while maintaining a focus on the determination of requirements as opposed to design activities. They argue that the inability to define contextual issues can lead to requirements which are incomplete, not verifiable, unnecessary, and unusable. Furthermore, placing emphasis on design activities too early in the lifecycle prioritises the goals of the developers above the needs of the users and may contribute to poor requirements. Christel and Kang (1992) therefore recommend that requirements elicitation begins with an organisational and context analysis to determine the boundary of the target system as well as the objectives of the system. They state that failure to do so may result in the production of requirements which are incomplete and potentially unusable, because they are not based on the user's or organisation's real needs, and add that performing an organisational and context analysis allows the needs of the users or organization to be effectively captured, and later verified.

ii. Problems of Understanding

Errors in installed systems are costly to correct and can often be traced back to poor communication between user and analyst in defining requirements (Goodrich, 1990) in Christel and Kang (1992).

According to Christel and Kang (1992) miscommunication between analysts and stakeholders can result in requirements which are ambiguous, incomplete, inconsistent, and even incorrect because they do not reflect the real needs of the stakeholders or the perception of the analysts.

Christel and Kang (1992) recognise that the various stakeholders involved in elicitation may be from a wide variety of backgrounds and experience levels, and hence the possibility exists that what can be defined as common knowledge to the analysts may be completely foreign to the other stakeholders. This complicates the process of interpretation and integration of information gathered from the diverse communities. They also acknowledge that communication between analyst and stakeholders is further challenged by the formality of the language used to express the analyst's interpretation of the requirements to the users, and the manner in which large volumes of information are modeled for presentation to the stakeholders.

They therefore recommend that requirements are expressed in a manner that promotes communication and understanding between customers and users; allows the developer to determine whether the expressed requirements are implementable; and affords quality assurance teams the ability to verify that the requirements have been adequately addressed.

iii. Problems of Volatility

Requirements elicitation activities need to accommodate the ability of requirements to change at any point of the development lifecycle. These changes may occur as a result of the changing needs of the users due to their increased knowledge brought on by the development activities, or the shift in needs due to unforeseen organisational or environmental pressures (Christel and Kang, 1992). They argue that failure to accommodate the evolving needs of an organization or its users may contribute to the incompleteness, inconsistency to the new situation, and potential lack of usability of the requirements as they are based on information that has since become obsolete.

4.4.3.1 A Model to Address Problems in Scope, Understanding and Volatility (Christel and Kang, 1992)

Christel and Kang's (1992) Requirements Elicitation Process model is based on the need for enhanced communication between the analyst and organisational stakeholders, as well as the

recognition of the possible diversity of backgrounds of the analyst and stakeholders. Christel and Kang (1992) describe the relationship between requirements elicitation and software design as symbiotic, with the initial set of requirements targeted at the initiation of the design process and gradually refined into a systematic and coherent statement of requirements simultaneously with the refinement of design.

The Requirements Elicitation Process model is composed of five categories of tasks: Fact Finding, Requirements gathering and classification, Evaluation and rationalisation, Prioritisation, Validation and integration. These tasks are further classified into two categories: user oriented tasks and developer oriented tasks.

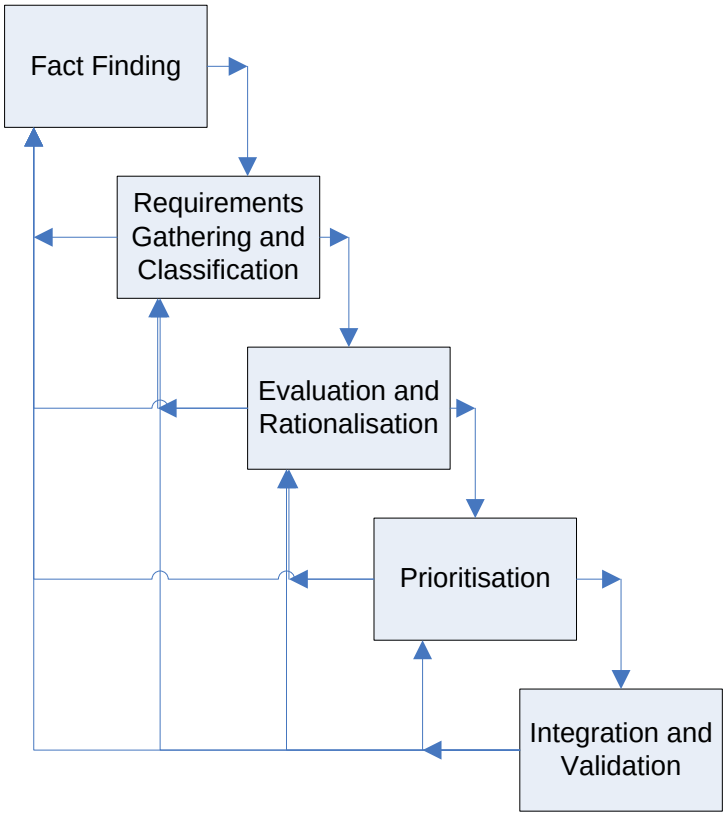


Figure 8: Requirements elicitation tasks according to the Requirements Elicitation Process Model (Christel and Kang, 1992)

Table 3: Requirements elicitation tasks according to the Requirements Elicitation Process Model (Christel and Kang, 1992)

User Oriented Tasks	Developer Oriented Tasks
1. Fact Finding – Identify Stakeholders, Operational and problem context, and similar systems – Perform context analysis	1. Fact Finding – Identify domain experts as well as domain and architectural models – Conduct technological surveys and assess cost and implementation constraints
2. Requirements Gathering and Classification – Get wish list for all parties across multiple levels	2. Requirements Gathering and Classification – Classify wish list according to functional, non functional, environmental and design constraints, as well as according to domain models (i.e. Top down functional decomposition or Object-Oriented Development)
3. Rationalisation and Evaluation – Evaluate and capture rationale for the inclusion of specific requirements	3. Rationalisation and Evaluation – Perform risk assessment (i.e. technical, cost and schedule)
4. Prioritisation – Determine criticality of included requirements	4. Prioritisation – Prioritise requirements based on cost and dependency – Evaluate potential for incremental development and explore architectural models that support incremental development
5. Integration and Validation – Address completeness by defining issues that have not yet been fully determined – Ensure that there is alignment between the requirements an the original goals – Sign off requirements before proceeding to next stage	5. Integration and Validation – Perform consistency checks to resolve conflicts

4.4.3.2 Strengths of the Requirements Elicitation Process Model

As opposed to The Task Characteristics Model (Browne and Rogich, 2001), and the Requirements Elicitation Technique Selection Model (Davies and Hickey, 2002:2004), Christel and Kang's Requirements Elicitation Process Model (1992) offers more flexibility as it does not prescribe specific techniques for requirements elicitation, nor does it explicitly list the factors that affect the technique selection process. Instead the model defines the problems associated with requirements elicitation (Problems of scope, understanding and volatility) and prescribes the activities necessary to overcome these challenges. The activities are structured such that the contextual factors are defined first, and therefore subsequent activities will take these factors into consideration. The model is thus context independent and is likely to accommodate software development in environments such as Dwesa.

Furthermore, in situations where analysts and users come from vastly contrasting backgrounds, such as in the case of the project in Dwesa, the model enforces a strong user orientation by viewing the requirements elicitation activities bilaterally from the perspectives of the users' and developers. This approach is likely to promote the discipline of encouraging the active involvement of users throughout the software development process, and hence enhance the sense of ownership of the software by the users.

Finally, the model ensures that the requirements are consistent with the original goals by continually monitoring and validating the requirements and adjusting them accordingly. This encourages the definition of a quality set of requirements by eliminating unnecessary requirements, clarifying vague or ambiguous requirements, and introducing previously omitted requirements based on the continuous evaluation of requirements throughout the process.

4.4.3.3 Weaknesses of the Requirements Elicitation Process Model

Although the context independent nature of the model offers it the flexibility to be adapted to a broader range of projects, it comes at the expense of the ability to address problems such as requirements techniques selection and application. Furthermore, the model is weighted more on the validation and management of requirements, and asserts little emphasis on how the requirements are obtained. The model is also vague in its description of how it interacts with the software development lifecycle.

4.4.4 The Matrix of Method Selection (Zhang, 2007)

Zhang (2007) describes the process of requirements elicitation as an intensive process of interaction between analyst and stakeholders and therefore recommends that requirements elicitation techniques be classified according to the manner in which they support this interaction. Four categories of techniques are described in Zhang (2007): Conversational, Synthetic, Observational and Analytic. Zhang (2007) adds that requirements elicitation technique selection is influenced by the level of abstraction of the requirements, communication obstacles and uncertainty levels.

i. Conversational Techniques

According to Zhang (2007), conversational techniques (also known as verbal methods) include techniques that make use of verbally expressed natural language communication between two or more people. Zhang states that such techniques are effective in developing an understanding of the problem and defining generic requirements. Conversational techniques include interviews, workshops, focus groups and brainstorming sessions.

ii. Observational Techniques

Observational techniques, according to Zhang (2007), allow the analysts to gain an understanding of the application domain by observing human activities. Such techniques are more successful in revealing tacit requirements that may be apparent to the stakeholders but difficult to articulate. Observational techniques include techniques such as direct observation, cognitive walkthroughs, ethnographic studies, and protocol analysis.

iii. Analytic Techniques

Analytic techniques aim at determining requirements based on deductions founded on expert knowledge or existing and documented domain knowledge (Zhang, 2007). This category includes techniques such as requirements reuse, document evaluation, card sorting, laddering and repertory grids.

iv. Synthetic Techniques

Zhang Supports the view that the process of requirements elicitation cannot be supported with a single technique, and that the techniques should be selected based on the context and circumstances surrounding the activities. Synthetic techniques combine conversation, observation, and analysis into single techniques that analysts and stakeholders can apply to communicate and interact amongst each other to reach a common understanding of the desired product. Synthetic techniques are also known as collaborative methods, and include techniques such as scenarios, passive storyboards, prototyping, interactive storyboards, Joint Application Development/Rapid Application Development and Use Case Modelling.

4.4.4.1 Other factors affecting technique selection

Zhang (2007) further categorises the process of requirements elicitation according to two levels of abstraction: problem analysis and product specification. According to her categorisation, problem analysis aims at defining the problem domain by understanding the situation of concern and setting boundaries, while product specification focuses on specific product features. These two classifications serve different goals and therefore engage different techniques to elicit requirements. Zhang also recognizes the influence of communication obstacles, the source of the requirements and the levels of uncertainty on the requirements elicitation processes. Zhang's (2007) categorization of techniques guides the technique selection process by defining the conditions under which the various categories of techniques can be effective.

4.4.4.2 Strengths of Zhang's Matrix of Technique Selection (2007)

Zhang's (2007) Matrix of Technique Selection addresses the main shortcomings of Browne and Rogich's (2001) Task Characteristics Model as it recognises that requirements elicitation techniques cannot be prescribed in the absence of an evaluation of the context of their application. Furthermore, as opposed to Davies and Hickey's Technique Selection Model, Zhang (2007) provides a broader more in depth description of the variables that affect requirements elicitation technique selection. The matrix also provides more enhanced guidance for technique selection as it categorises the techniques according to the manner in which the requirements are communicated between the analysts and the various stakeholders or sources. It therefore provides analysts with a variety of techniques that can be applied based on the respective scenarios. The matrix also rates the categories in terms of their effectiveness under each relevant circumstance. It is therefore a more practical guide to requirements elicitation than the other models discussed in this chapter.

4.4.4.3 Weaknesses of Zhang's Matrix of Technique Selection (2007)

Although the Matrix of Technique Selection is founded on empirical and theoretical findings of cognitive psychology, it is based on secondary evidence and a likelihood therefore exists that a certain amount of subjectivity exists in the ratings of the various categories of techniques under the various circumstances. Zhang (2007) acknowledges this fact and states that the degree of applicability of each method has to be further adjusted and customised in different organisations according to their work settings and the nature of the project. The ratings serve merely as a guideline rather than a rule.

Table 4: Zhang's (2007) Matrix of Technique Selection

KEY:
√√√: Methods strongly recognize the issue and provide a means to deal with it
√√: Methods support the issues, but not very strongly
√: Methods address the issues, but weakly or indirectly
×: Methods do not address the issues.

Perspective		**Conversational**	**Observational**	**Analytic**	**Synthetic**
Abstraction level	Problem Analysis	√√	√√√	√√	√
	Product Description	√√	√	√√	√√√
Requirements Source	Human beings	√√√	√	√	√√
	Other Environments	×	√√√	√√	×
Communication Obstacles	National Culture	√√	√	√	√√√
	Organisational Culture	√	√√√	√√	√
	Cognitive Limitations	√√	√	√	√√
	Geographically distributed environments	√√	×	√	√
Level of certainty	Existing Domain	√√√	√	√√√	√√
	New Domain	√√	√√	√	√√√

Conclusion

Over the decades software development methodologies have evolved to address the inherently complex nature of software development. The introduction of 4GL's, OOSE and CASE has further simplified the process of software development and shifted the focus from targeting complexity to ensuring that the needs of the users have been met. The greatest challenge in addressing the needs of the user is that of determining exactly what their needs are. Successful requirements elicitation is very critical to the overall success of any software development project. Unfortunately it has been largely neglected in research and application.

The previous chapter evaluated requirements elicitation techniques specifically, while this chapter compares the suitability of various requirements elicitation models to a rural context. Four models were evaluated: Browne and Rogich's Task Characteristics Model (2001), Davies and Hickey's Techniques Selection model (2002) Christel and Kang's Requirements Elicitation Process Model (1992) and Zhang's (2007) Matric of Technique selection. The comparison revealed that both the Task Characteristics Model and the Techniques Selection Model's strengths exist in the manner in which they recognise the relationship between the problem and solution domains, the stakeholders, the project/organisation, and the elicitation process. The main difference between the models is that Browne and Rogich (2001) recommended the use of specific prompting techniques to interface the relationship, while Davies and Hickey (2002) recommended a technique selection process to determine how to interface the relationship. The main criticism of the models were that one model was too rigid as it prescribed specific methods for eliciting user requirements, while the other model's rigidity was in its' diagnosis of the factors that determine the requirements elicitation techniques to be applied.

Christel and Kang's (1992) model addresses these shortcomings by presenting a model that neither prescribes the use of specific techniques, nor attempts to identify the variables affecting technique selection. Unfortunately, it fails to elaborate on how the initial requirements are obtained, as its main emphasis is on managing the requirements as well as enhancing communication between analysts and users.

Zhang's (2007) Matrix of Technique Selection is flexible as it does not prescribe specific techniques to apply, and although it identifies the factors that influence the technique selection process, these factors are expressed as broad categories rather than individual variables. Furthermore, Zhang's (2007) approach is more practical and specific as it guides the analyst in terms of the suitability of the techniques in addressing various issues.

All the approaches evaluated in this chapter have their merits and shortcomings. The aim is not to criticize them, but rather to draw from their strengths in order to develop an approach to requirements elicitation that can be effectively applied in a rural scenario. The following chapter presents a model for requirements elicitation that combines the strengths of the reviewed approaches into an improved model that is specifically suited to rural ICT implementations.

Chapter 5

A Model for Requirements Elicitation Suitable for Software Development in Rural Areas

5.1 Introduction

A review of the effectiveness of various requirements elicitation techniques within a rural context was conducted and revealed that it is insufficient to evaluate requirements elicitation techniques without understanding the context of their application. A variety of requirements elicitation models were subsequently evaluated. Browne and Rogich's Task Characteristics Model (2001) and Davies and Hickey's Techniques Selection model (2002) both model the interaction between the problem and solution domains, the stakeholders, the project/organisation, and the elicitation process appropriately, but provide rigid means of addressing the challenges in their interaction.

Christel and Kang's (1992) model provides more flexibility as it neither prescribes the use of specific techniques, nor attempts to identify the variables affecting technique selection. Unfortunately, it also fails to elaborate on how the initial requirements are to be obtained, as its main emphasis is on managing the requirements and enhancing communication between analysts and users. Zhang (2007) provides a more specific and practical approach to requirements elicitation that supports the process of determining the techniques that can be applied under various circumstances. It therefore addresses the main shortcomings of Christel and Kang's (1992) Model. However Zhang's (2007) model's flaws stem from the level of subjectivity present in the manner in which it rates the categories of techniques according to their ability to address certain problems.

This chapter presents a model that aims at merging the strengths of the discussed approaches, and minimising or eliminating their shortcomings. This will be achieved firstly, by compiling a summary of the core elements from which the proposed model will be drawn, and secondly, by illustrating the manner in which these elements interact to support the requirements elicitation process.

5.2 Elements Drawn from Existing Models

Based on the evaluation of the Task Characteristics Model, The Requirements Elicitation Technique Selection Model, and The Requirements Elicitation Process Model, and the Matrix of Technique Selection, various elements were of particular significance. Firstly, the manner in which Browne and Rogich (2001) describe the interaction between users, analysts and their organisation, and the challenges of the analyst in eliciting the user's requirements, provided a high level view of the process of requirements elicitation and the challenges that must be overcome in order to enhance the analyst's ability to shift the organization from its current state to an improved state. Davies and Hickey's model (2002) further introduced the idea that prior to Requirements elicitation, a technique selection process is necessary based on the variables surrounding the problem and the project. Finally, Christel and Kang (1992), categorsied the activities involved in requirements elicitation into two classes: User Oriented and Developer Oriented, while Zhang (2007) categorized the techniques according to the manner in which they support interaction between analysts and users, and described how the techniques are affected by various elements.

These elements will be combined into a methodology that can support requirements elicitation in a rural context. A model that categorises the elements that affect the requirements elicitation process as well as the technique selection process is presented in this section to address the need for a requirements elicitation methodology for rural software development.

5.3 Rationale For The Categorisation

Requirements elicitation is often wrongfully assumed to be limited to the analysis phase of the SDLC (Zhang, 2007). Subsequent requirements activities in later stages of development are hence categorised as requirements expression and validation techniques (Rzepka, 1989), and often emphasise the representation of the requirements determined in the analysis phase, rather than attempting to refine and improve on the quality or quantity of real requirements identified. User requirements elicitation is an ongoing process and is not limited to the analysis phase of the SDLC. This view is supported by both Christel and Kang (1992) and Zhang (2007). Technique selection should thus be recognised as an ongoing process as well in order to sustain the analysts focus on the application of appropriate techniques throughout the SDLC.

Table 5: An example of elicitation methods applicable at various stages of software development and the type of information they yield

Stage of Development	Requirements Sought	Examples of applicable Techniques
Analysis	Requirements that reflect the current challenges and needs of the organisation and users	• Interviews, JAD, Focus Groups, Needs Analysis
Specification	Requirements that reflect a unified understanding of the organisation, users and system needs and goals, as well as requirements that have been omitted or poorly defined during the analysis phase	• Use Case Models • Data Flow Diagrams • Sequence Diagrams • Scenarios • Collaboration Diagrams • Lo-Fidelity prototyping
Design	Misinterpreted, misrepresented or omitted requirements, as well as requirements that are mismatched with the architectural and domain models in use	• Low Fidelity Prototyping • High Fidelity prototyping
Implementation	Requirements that have been poorly executed, are unnecessary or have been omitted during earlier phases	• Cognitive Walkthroughs • Direct Observation • Heuristic, guideline and Expert Reviews • Formal Inspection
Operation	Requirements that have been poorly executed, are unnecessary or have been omitted throughout the lifecycle	• Direct Observation • Acceptance tests • Surveys • Expert Reviews • Formal Inspection • Continuous assessment during active use

At various stages of the Software Development Lifecycle (SDLC), different types of knowledge or information are required by the analyst regarding the system to be developed. The depth and detail of information required from the various stakeholders varies throughout the lifecycle and is determined by the goals of the phase of software development. Zhang (2007) also recognises the

distinction between the categories of knowledge pursued by analyst and distinguishes them into two phases: problem analysis and product description.

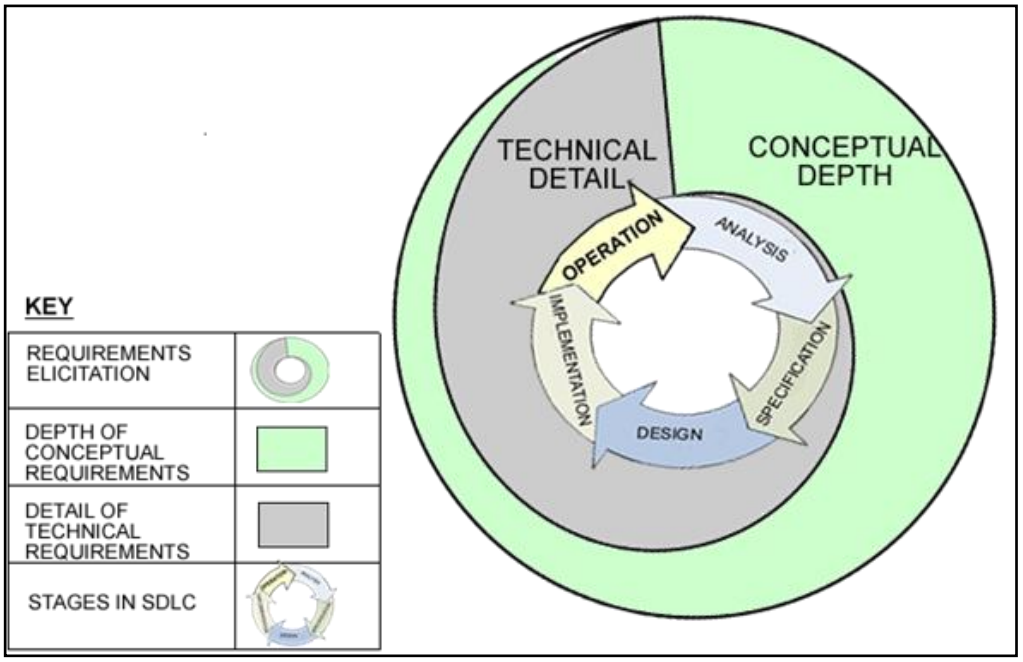


Figure 9: An illustration of the change in information required during the SDLC

According to Zhang's Matrix of Technique Selection, the selection of elicitation techniques based on their classification (conversational, observational, synthetic and analytic) is influenced by the level of abstraction, communication obstacles and the degree of uncertainty. The proposed model will classify the level of abstraction into two layers: the conceptual layer and the technical layer. The model will further classify the communication obstacles according to the analyst's perspective and the user's perspective of the problem domain. Finally the model will distinguish the requirements elicitation activities based on the stage of software development in which they are implemented. As the proposed model is already being defined to accommodate requirements elicitation in environments where there is a high degree of uncertainty, the model will not make any explicit reference to the effect of the degree of uncertainty.

5.3.1 Conceptual Depth

The initial interaction with the client should aim at identifying the problem and creating a case for the project. Zhang (2007) describes this process as problem analysis. This point is described by the researcher as one of maximum conceptual depth and minimal technical detail and is characterised by the application of strategies to obtain an in-depth understanding of the various elements surrounding

the environment within which the system is intended to be implemented. This includes the knowledge of the prospective users, the existent business processes, socio-political structures, the problems surrounding the targeted community, the skills available amongst the development community, the time constraints, and the resources available. The list of variables affecting the attainment of high conceptual depth requirements is endless, and the depth required varies from project to project. During this stage the analyst must select appropriate requirements elicitation techniques based on the variables that affect elicitation in the specific project. It is rarely possible or even desirable to allocate strict time deadlines to this stage as it is largely dependent on the response of the stakeholders. If executed effectively this stage of requirements elicitation should result in the achievement of stakeholder's goals, user satisfaction and the ultimate improvement or solution of the problem.

5.3.2 Technical Detail

As the Project progresses into subsequent stages of the SDLC, the level of conceptual depth required from the elicitation process will decline, while the level of technical detail necessary will increase. At this stage there is an increasing need to translate the requirements into an end product. This process is described by Zhang (2007) as product description, and is characterized by the diminishing need for knowledge regarding the problem domain, and the rising need for definition of the solution. This stage is characterised by an increasing dependency on the knowledge, skills and expertise of analysts, decreasing dependency on interaction with stakeholders, predictable time frames for completion, and increasing detail in terms of the technical system requirements. The effective implementation of this stage will determine the overall quality of the end product, the level of usability and the efficiency of the software.

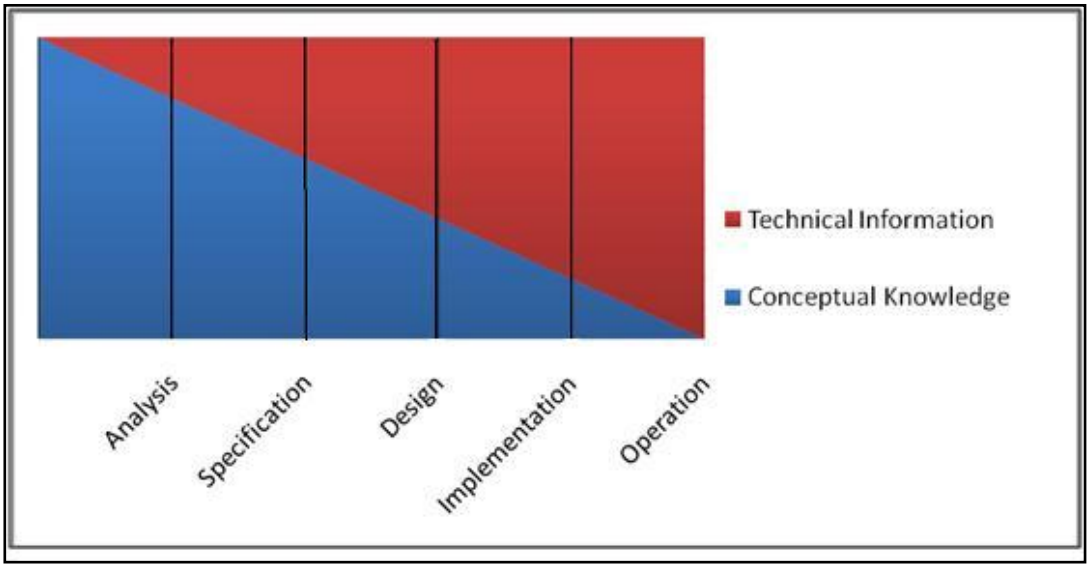


Figure 10: The Relationship between Conceptual and Technical Requirements

Although the trend exhibited above appears as a straight linear relationship between conceptual and technical requirements, it is likely to be less predictable and would probably be categorised by an irregular pattern. The end of the operation phase is also likely to be characterised by an increase rather than a decrease in the proportion of conceptual requirements as the focus of the software developers will begin to shift back to the users needs as they evaluate the degree to which the users needs have been met. Furthermore, the operation phase informs future versions, increments or enhancements of the software and is therefore likely to require conceptual information that will be taken into consideration during the analysis phase of the future versions. Nevertheless, the proportion of conceptual requirements obtained in relation to technical requirements is still likely to exhibit an overall decline between the analysis phase and the operation phase of development.

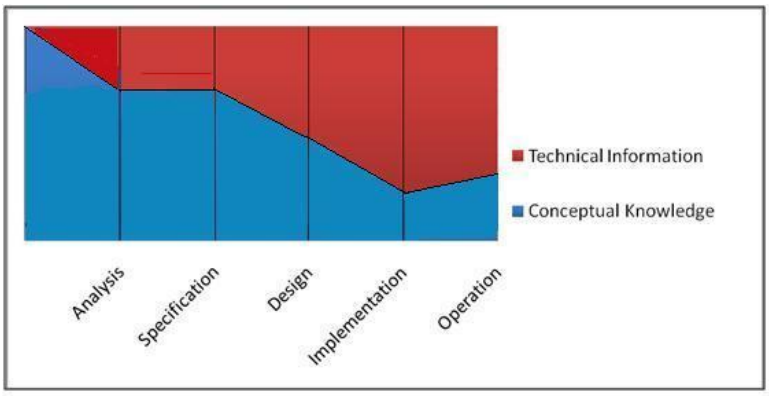


Figure 11: Realistic depiction of the relationship between conceptual and technical requirements

5.3.3 Categorisation According to Nature and Degree of Interaction between Analysts and Users

As stated in the previous chapter, problems in understanding results in costly errors that occur as a result of poor communication between users and analysts in defining requirements (Goodrich, 1990) in Christel and Kang (1992). This view is supported by Browne and Rogich (2001), who identify the inability of the users to articulate their views as one of the challenges in requirements elicitation. Furthermore, Christel and Kang (1992) recognise that diversity in the backgrounds and experience levels of the various stakeholders complicates the process of interpretation and integration of information gathered. Christel and Kang (1992) also recognise that communication between analyst and stakeholders is challenged by the formality of the language used to express the analyst's interpretation of the requirements to the users, and the manner in which information is modeled and presentation to the stakeholders. They further state that requirements should be expressed in a manner that promotes communication and understanding between stakeholders; allows analysts to determine whether the expressed requirements are implementable; and affords quality assurance teams the ability to verify that the requirements have been adequately addressed. The enhanced communication between analyst and users is critical to the success of the requirements elicitation process and must be effectively addressed in requirements elicitation methodologies. Technique selection is further influenced by the degree and nature of interaction between the analysts and users during the specific stages of development.

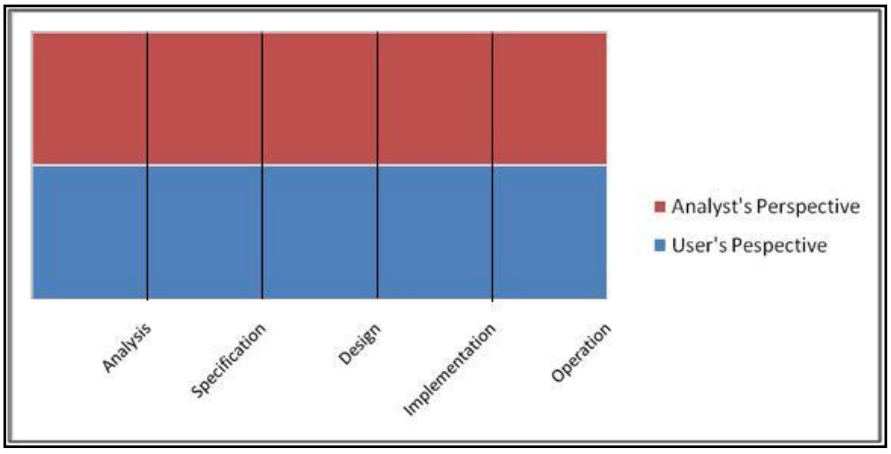


Figure 12: Relationship between User and Analyst Domain

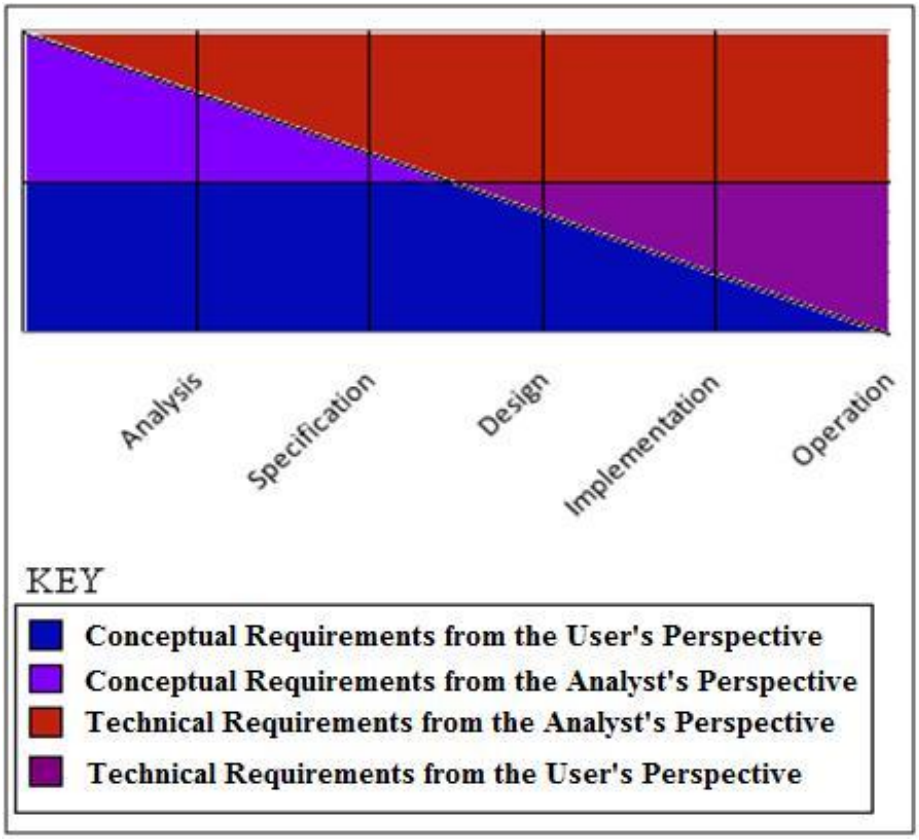


Figure 13: The Interaction between the User's Domain, Analyst's Domain, Conceptual Requirements, and Technical Requirements

5.3.4 Methods to Obtain Conceptual Knowledge from the Users Domain

i. Conversational Techniques

The proposed model recommends the application of conversational methods to elicit conceptual requirements from the user's domain as they provide a natural form of expression of needs and ideas enabled through conversation (Zhang, 2007), and is the primary and most prevalent means of acquiring non-tacit knowledge (Maiden and Rugg, 1996), knowledge which according to Polanyi (1967), can be easily codified, transmitted and published and does not vary from person to person. Maiden and Rugg (1996) add that people that are willing to recall and communicate knowledge tend to do so verbally and support the view that non tacit knowledge is best acquired verbally.

5.3.5 Obtaining Conceptual Knowledge based on the Analyst's perspective of the users Domain, and Technical Information from the Users Perspective

i. Synthetic Techniques

According to Polanyi (1967), tacit knowledge refers to poorly defined, unpublished, un-codified or subjective knowledge which can not be fully expressed and exists amongst people sharing a common experience. Smith (2003) describes tacit knowledge as the conceptual and sensory information that results from the efforts to make sense of something. Polanyi (1967) adds that the act of gaining knowledge exercises a personal judgement in relating evidence to reality. According to Zhang (2007), synthetic elicitation techniques combine the application of various communication channels and make use of rich semantic models to support the identification of requirements and allow analysts to validate their view of the users' problem space with the users themselves. The proposed model therefore recommends the use of synthetic requirements elicitation techniques to merge the users' and analysts' view of the proposed system. The application of synthetic techniques to gain conceptual knowledge about the user's domain from an analyst's perspective will allow the analyst to model their understanding of the current situation for validation with the user. The application of methods to merge the views of the analysts and users is a critical but often neglected element in the process of requirements elicitation for rural software projects. This view is supported by Heeks (2002), who states that the design conceptions that inform the development of software are often based on the assumptions of the stakeholders who dominate the process. He adds that stakeholders often base the design on the contexts in which their prior experience has been founded and hence a gap is likely to exist between the system's design and the reality of its users.

ii. Observational Techniques

Stakeholders are not the only source of knowledge in the requirements elicitation process (Maiden and Rugg, 1996; Zhang, 2007). Knowledge can also be found in domain objects such as physical artifacts (Bright et al., 1995) in Maiden and Rugg (1996), as well as abstract entities such as social networks (Yu, 1993) in Maiden and Rugg (1996). Maiden and Rugg (1996) and Zhang (2007) agree that observational techniques are effective in eliciting such knowledge. Maiden and Rugg (1996) add that implicit knowledge, which is, according to Seger (1994), knowledge that is acquired

subconsciously, and compiled knowledge, which exists as a result of extensive practice and habit is not expressed through verbal channels. They state that observational techniques are the best means of acquiring such knowledge. Observational techniques therefore support the process of eliciting conceptual knowledge from the user' s domain from the analysts perspective, as it allows the analyst to observe phenomena that have not been articulated verbally. Furthermore, observational techniques can also be applied to evaluate how the user interacts with the software system during the design, implementation and operation phases of software development.

5.3.6 Determining Technical Requirements Based on the Analyst's Perspective

Analytic techniques, according to Zhang (2007) engage the use of experts' knowledge for the critical evaluation of documentation, or the determination of requirements through a series of deductions. This form of knowledge is processed mainly in the analyst's domain and is founded on tested and documented evidence. Such techniques are effective in determining technical requirements based on the analyst's perspective.

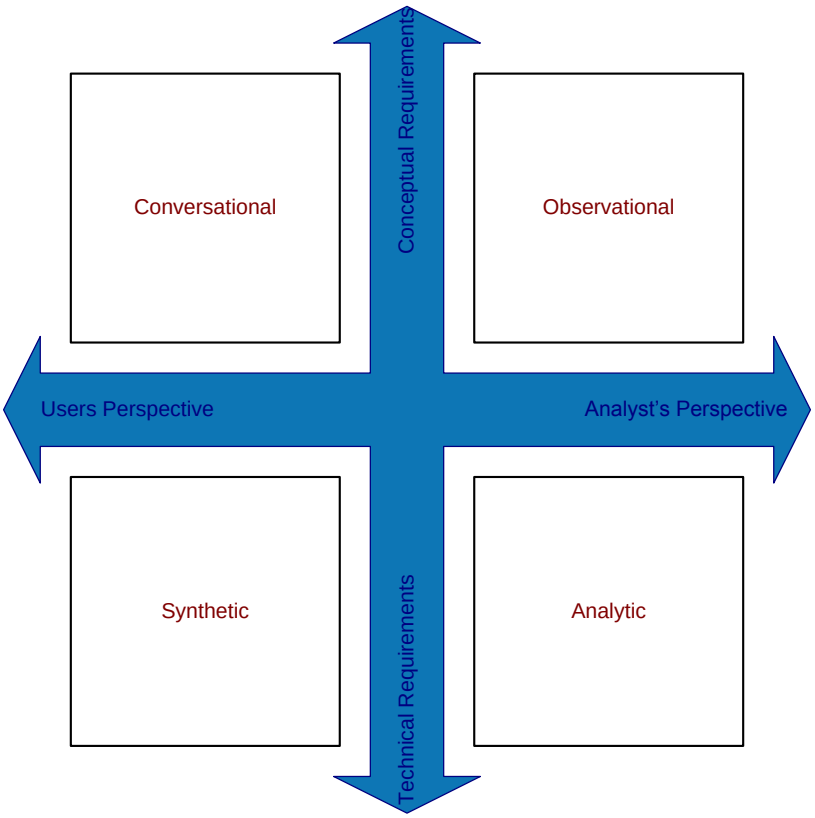


Figure 14: An illustration of how categories of techniques apply to various conditions

5.4 Conclusion

The proposed model combines the strengths of various models to enhance the process of requirements elicitation by informing the technique selection process. It categorises the elements of requirements elicitation according to the knowledge or information sought, the stage of the SDLC within which they fall, the perspective from which they are viewed (analyst's or user's), and the level of abstraction (conceptual depth or technical detail). Based on these categories it applies theoretical and empirical research to recommend a class of applicable requirements elicitation techniques.

According to the model, conversational techniques are the most effective means of gaining conceptual knowledge from the user's perspective. Observational techniques can be effective in providing the analyst with an insight into the conceptual domain, while synthetic techniques can be applied to communicate the analyst's view of the conceptual knowledge for validation with the users. To determine the degree to which the user's needs have been addressed, technical information is required from the user's perspective. As this type of information is difficult to articulate for users with limited experience with such technologies, observational techniques are best suited to gain such information. Finally, when determining requirements based on the analyst's understanding of technical information, less user involvement is required as this type of information is founded on the analysts' knowledge of their discipline. Analytic techniques are the most useful under such circumstances. In order to validate the model, it is necessary to evaluate it in terms of an actual rural ICT project. The chapter that follows will explore the model further in the context of the Siyakhula Living Labs Project.

Part 3: Empirical Framework

Chapter 6

Research Design

6.1 Research Paradigm

This research will be founded on interpretivist approach to the evaluation of the process of requirements elicitation in Dwesa. Interpretivism, according to O'Brien (2001) is influenced by culture and history and founded on the belief in socially constructed, subjectively based reality. Interpretivism, as opposed to positivism research, does not aim at proving or disprove a theory, but rather at identifying and understanding the interactions and interrelationships within a given setting (Oates, 2006). According to Oates (2006) Interpretivist studies are founded on the belief that reality does not exist as a single truth but is rather perceived as multiple subjective realities. These realities are accessed or transmitted through dynamic social constructs such as language, shared meanings, and understanding. Oates (2006) adds that researchers are not neutral in the process and therefore their assumptions, beliefs and values will inevitably influence the research process. Interpretivist research does not aim at arriving at a single explanation of a phenomenon, but rather explores various explanations, while endorsing the strongest (Oates, 2006).

6.2 Underlying Theory

The research is founded on Critical Social Theory which focuses on the inequalities in society and is believed to be constituted in history and reproduced by people (Avison and Myers, 2005). Critical Social Theory seeks to minimise or eliminate the causes of marginalization and domination of historically oppressed groups. With regard to this research, this marginalisation refers to the divide in access and use of Information and Communication Technologies to gain access to information, service and markets for social and economic upliftment. This marginalisation is a result of over a century of social injustices imposed on the residents of Dwesa that has consequently led to the low levels of literacy and access to goods and services by the community.

This research project also draws from the Technology Acceptance Model (Davis, 1986:1989) which recognises the influence of the perceived ease of use, as well as the perceived usefulness on the behavioral intention to use the technology and ultimately determine the actual use of the technology.

User requirements elicitation aims at understanding what the needs of the users are in order to develop applications that can be perceived as useful. As this study aims at reviewing the process of requirements elicitation for the purpose of enhancing the adoption and sustained use of software applications in rural communities, it bases many of its assumptions surrounding user behavior on this theory.

6.3 Research Strategy

Although software development issues have been broadly researched and well documented, the understanding of their application in a rural context is limited, and the measure of their success or failure is subjective (Van Belle and Trusler, 2005) in Pade (2006). Nevertheless, it is generally accepted that project failure tends to be more prevalent than project success in rural areas (Pade, 2006; Heeks, 2002). There is therefore a need for an increased understanding of the elements that affect the effective and sustained use of technology in a rural setting.

Triangulation involves the application of two or more strategies to address the research problem and provides a broader more complimentary view of the problem (Oates, 2006). Collis and Hussey (2003) claim that such approaches help researchers to overcome the subjectivity of single strategy. This project will follow a strategy triangulation process to combine the case study and participatory action research approaches to answer the research questions. The triangulation approach will help minimise the subjectivity often associated with qualitative studies. As the case study (The Siyakhula Living Labs Project) engages Participatory Action Research (PAR) to achieve its objectives, it is inevitable that most of the principles of this approach will manifest in the study, particularly in the evaluation of requirements elicitation activities.

6.4 The Participatory Action Research And Case Study Methods

Yin (2003) describes case studies as empirical enquiries into contemporary phenomena within their real context. Using the Siyakhula Living Labs Project as a case study will allow the researcher to observe, evaluate and report on multiple aspects of the requirements elicitation activities within a

rural context. Yin (2003) states that case studies are a useful method of research as they provide real-life information and evidence.

Applying a PAR approach in the actual examination of the requirements elicitation techniques will further add to the validity of the findings. Action Research, according to Lewin (1946), involves the investigation into the conditions and consequences of various forms of social action and involves planning, action, and fact finding about the result of the action. The community centeredness of PAR makes it highly relevant to this study as it allows one to obtain knowledge based on her experiences in interacting with the community within which the E-Commerce platform will be deployed.

Various definitions of Participatory Action Research (PAR) are provided in Collins (1999). Firstly Collins herself defines PAR as the collective generation of knowledge resulting in the planning and achievement of jointly set goals. Collins adds that the objectives of PAR often serve political ends, but are sometimes driven by the need for organisational change, project management, community development or personal growth. Hall (1983) in Collins (1999) defines PAR as a process that combines research, education and action. Fals-Borda and Rahman (1991) expand on that definition by describing PAR as a combination of three elements; research, education and socio-political action. They elaborate that these elements can be combined into an experiential methodology made up of collective and personal behaviour, that implies the acquisition of serious and reliable knowledge upon which power or countervailing power can be constructed for the disempowered or disadvantaged. According to Fals-Borda and Rahman (1991), the knowledge resulting from PAR serves two aims; firstly, to enable oppressed groups to acquire sufficient creative and transforming leverage through projects, acts and struggles, and secondly, to produce and develop socio-political thought processes with which popular bases can identify. Mulenga (1994) in Collins (1999) similarly view PAR as an emancipatory approach to knowledge production and utilisation that aims at investigating and transforming the reality of the disenfranchised.

Tandon (1988) in Collins (1999) explains that PAR attempts to present people as researchers in pursuit of answers to their daily struggles, and Heron (1996), in Collins (1999), describes PAR as person-centered research that conducts research with people as opposed to conducting research on them. Collins definition of PAR is accurate as it does not restrict the goal of PAR to empowerment

or development, but acknowledges the role it can play in organizational change, project management and even personal growth. The importance of Heron (1996) and Tandon's (1988) definition of PAR with respect to the Siyakhula Living Labs Project is their emphasis on the people centeredness of the approach, and the fact that the community members themselves are perceived as part of the research team.

PAR, as described in Mouton (2001) is characterised by the involvement of the subjects of the research as an integral part of the design. According to Mouton (2001), it usually engages the use of qualitative methods to gain an insight into the lives of the research subjects, and in many cases there is an explicit commitment to the empowerment of the participants. Mouton (2001) adds that successful PAR studies are likely to result in a high level of involvement by the participants and hence, has the ability to enhance the chances of a high construct validity, low refusal rates and high ownership of the findings. The main weakness however, according to Mouton (2001), is that, as is typical in interpretivist studies, the researchers have the ability to affect the research process to serve their own interests. The author will however, strive to maintain a high level of objectivity as a passive elicitor and interpreter of information as prescribed by O'Brien (2001).

Based on the above descriptions, this research defines PAR as a practical people centered approach to gaining knowledge and improving conditions within communities that encourages the opinions and efforts of all the stakeholders. Furthermore, although the knowledge acquired through PAR has the potential to empower the community from which it is derived, the knowledge gained through PAR can also make a difference beyond the boundaries of the community within which the research has been conducted.

In a situation where technology is being deployed in less understood environments, it is critical that analysts attempt to understand the context in which the technology will be applied. This requires interaction with the community in which the technology will be applied. As this study forms part of a larger ICT project, the high level of cooperation between the developers and the prospective users that results from the PAR methodology in the development of the E-commerce platform will be highly advantageous. PAR allows the developers to become actively involved with the community and allows them to perceive the system requirements from the community's perspective.

6.5 Research Methods

PAR does not follow any formal sequence of steps, but according to Atweh, Kemmis and Weeks (1998), it involves a spiral of self-reflective cycles of planning a change, acting and observing the process and sequence of the change, reflecting on the processes and consequences, and replanning. Atweh, Kemmis and Weeks (1998) explain that the stages of planning, acting and observing are not always sequential and may sometimes overlap. According to Atweh, Kemmis and Weeks (1998), the measures for success in PAR projects do not depend on the degree to which the participants have followed the steps, but rather on their dedication to development and evolution in their practices, their understandings of their practices, and the situation in which they practice. As the first stage of Participatory Action Research the (planning stage) has already been completed by the project initiators, this research will focus on acting, observing and reflecting on the process and consequences of the change. As opposed to replanning, which is the fourth stage in the PAR cycle of activities, this research will contribute by making recommendations for future research. The results of this study will be used to inform future projects within the project group.

6.5.1 Data Collection

i. Interviews

Oppenheim (1992) recommends the use of exploratory and standardised interviews at various stages of the research process. He elaborates that exploratory interviews serve to develop ideas and research theories rather than to gather facts or statistics. During the exploratory phase, the number of interviewees rarely exceeds forty people. The interviews aim at understanding how ordinary people think or feel about the subject thus, the interviewer should be sensitive to what the interviewee says, as well as to what is unspoken. The initial visit to Dwesa will aim at becoming familiar with the area and establishing links with the community. Subsequent visits, will initiate investigations of potential and existing entrepreneurial activities in the area through interaction with the local entrepreneurs. At the start of the project, very little is known concerning the nature of business activities in the area. Semi-structured exploratory interviews will be used as a primary means of enquiry. During the interviews the researchers will introduce the entrepreneurs to the concept of E-commerce, and help them explore ways in which their businesses can benefit from E-Commerce. According to Oppenheim (1992), the interviewing process is a highly complex one.

The interviewer should have sound interpersonal skills to allow the respondent to feel at ease and to promote the capturing of information without upsetting the flow of the conversation. Oppenheim also states that the process can be hindered by variables such as the age, gender, race and background of the interviewer. With this in mind, there is a need for a translator with whom the Dwesa residents can relate. One of the project members that is also from a semi-rural background, who speaks Xhosa (the language spoken in Dwesa), will assist with the translation.

ii. Observation

According to Maiden and Rugg (1996) and Zhang (2007) knowledge does not exist solely in stakeholders but can also be found in domain objects such as physical artifacts as well as abstract entities such as social networks. Maiden and Rugg (1996) and Zhang (2007) both concur that observational techniques are an ideal mechanism for eliciting such knowledge. Observation will be applied throughout the duration of the project to gain an understanding beyond what has been verbally expressed by the stakeholders. Furthermore, a prototype of the proposed system will be demonstrated in the course of the project. The researcher will be present to observe the prototyping process. The main objective of the observation will be to gain an insight as to the effectiveness of prototyping as a means of requirements elicitation in a rural context.

iii. Surveys

A survey will be conducted to evaluate the approaches to software development and in particular, requirements elicitation applied by the software developers. This will enable the assessment of the relationship between the software development and requirements elicitation activities, and the eventual attitude towards the technology being introduced. The survey will be composed of five multiple response questions. Each of the five questions will correspond to one stage of software development, and will require the respondent to select and rate the elicitation techniques they applied during the respective stage. The respondents will be required to rate each of the requirements elicitation techniques they applied into one of the following categories:

Table 6: Categories of ratings and their respective weighting

Category	Weighting
Completely conceptual	1.5
Mostly Conceptual	1
Equally Conceptual and Technical	0
Mostly Technical	1
Completely Technical	1.5

As the respondents all have internet access, and reside geographically dispersed, the survey will be distributed electronically. A pilot of the questionnaire will be distributed via e-mail. After making adjustments based on the feedback from the pilot, the questionnaire will be made available as an online survey. This will allow the results to be captured, analysed and communicated in real time from all the respondents regardless of their location.

iv. Documentary Evidence

Ongoing processes of reflection will take place throughout the duration of the project. Fortnightly progress meetings will be held amongst project members and an e-mail list will be used as a means of communication amongst the group. This communication will be documented when necessary to substantiate or validate observations. A review of literature in the field will be conducted to substantiate the findings

6.5.2 Sampling

Purposive sampling will be applied in the selection of the respondents of the study. Purposive sampling, also referred to as judgemental sampling, is a method commonly used in PAR studies. According to Collins (1999), this type of sampling involves allowing researchers to select the respondents based on their knowledge and experience. Data will be collected from two sources: the community members by means of exploratory interviews, and the four software developers by means of a survey.

6.6 Conclusion

This chapter describes the application of PAR and the case study method as a means of conducting interpretivist research on the Siyakhula Living Labs Project. PAR provides a community centred and therefore objective manner of understanding the variables affecting user requirements elicitation and is therefore well suited to research in less understood environments such as Dwesa. Semi-structured exploratory interviews with entrepreneurs in Dwesa will be conducted to identify Dwesa's current entrepreneurial activities and explore ways in which e-commerce can positively affect the community. A survey will be conducted amongst the software developers later to review the processes they applied to obtain the user requirements. Purposive sampling will allow the selection of a suitable group of project stakeholders to be interviewed. Observation will also be relied upon as a passive method of enquiry. The chapter that follows will consolidate and present the findings of the research activities.

Chapter 7

Eliciting User Requirements in The Siyakhula Living Labs Project: A Case Study

7.1 Introduction

Requirements elicitation is an inherently complex process that has not been sufficiently researched in the context of rural software development. Rural software initiatives face adverse challenges that stem from the socio economic conditions, historical injustices and marginalisation encountered by rural societies. In order for Information and Communication Technologies (ICTs) to address some of the problems associated with marginalised communities, their implementers need to understand the needs of such communities. To achieve this, a model that supports the elicitation of user requirements was presented in the previous chapter. This chapter serves to evaluate the model based on the requirements elicitation activities of the Siyakhula Living Labs Project based in Dwesa. The chapter introduces the project by elaborating on the chronological sequence of events that resulted in its conception before proceeding to describe the study.

An extensive analysis of the requirements elicitation activities of the software developers is provided in reference to the components of the proposed model in order to validate the potential value of the model. Analysis of the project's requirements elicitation activities revealed that the software developer's requirements elicitation activities correspond reasonably with the foundations of the proposed model. Furthermore, the findings enabled the researcher to make appropriate modifications to the model, based on her analysis, experiences and observations. The findings also allowed the researchers to identify the failure to interact effectively with the users when applying synthetic requirements elicitation techniques as the major flaw in the requirements elicitation activities of the software developers of the Siyakhula Living Labs Project.

7.2 Background

Dwesa is a rural town located in the Eastern Cape Province of South Africa. The past two decades have seen a rising interest in the challenges and achievements of the community. This interest stems from the community's rich political history which dates back to as early as the mid 1800s when the cattle killing movement (Fay, Timmermans and Palmer, 2002) that began in that region resulted in the deaths and displacement of large masses of inhabitants. Further adversities followed between 1936 and 1989, when the conservation laws denied the community the ability to reside in the forests, and from 1981, when they were further denied access to critical resources from the forest, grasslands and shoreline (Fay, Timmermans and Palmer, 2002). During these periods, an increasing portion of land was classified as part of a protected area and simultaneously reserved for "European recreation", while the indigenous inhabitants were being denied access to basic resources from the protected area (Fay, Timmermans and Palmer, 2002). The injustices were exacerbated by the drought in the early 1990s as the community's need for vital resources (food, water and firewood) in the protected area became more critical (Fay, Timmermans and Palmer, 2002). At this point the community had begun to exert increasing pressure on the administration to allow them access to the protected area. Their efforts attracted a great deal of media coverage and culminated in 2001, when they successfully claimed ownership of their land in what was only the second joint-management agreements to be signed in South African history in respect of a land restitution claim on a protected area .

Regaining ownership of land does not necessarily imply that the livelihood of communities will improve. This is evident in the case of Dwesa's land restitution success. Today, the area remains impoverished and critically underdeveloped and the residents are lacking in basic resources and services. Nevertheless, the activities in the area in the course of the century have captured the interest of a broad range of researchers. As a result, Dwesa was earmarked as a site to pilot a project to develop an E-Commerce platform to support rural development by the Telkom Centre of Excellence (CoE) for Developmental E-Commerce of Fort Hare and Rhodes University. These institutions have applied the efforts of a multi disciplinary group of post graduate research students to engage in research surrounding the provision and development of software, hardware, connectivity and training, as well as to investigate issues pertaining to the commercial and human

aspects of the initiative, the indigenous knowledge systems, and the delivery of public services through an E-Government portal.

7.3 The Siyakhula Living Labs Project

Ongoing anthropological research between 2001 and 2003 resulted in the establishment of a relationship between the residents of Dwesa and the Rhodes University Anthropology Department. This relationship enabled the conception of a research programme to experiment with the possible benefits of internet connectivity in rural communities (Muyingi, 2006). The Siyakhula project, which was later renamed the "Siyakhula Living Labs Project" in 2008, was initiated in 2006 as a joint initiative between the Telkom Centres of Excellence (CoEs) at Fort Hare and Rhodes Universities. As an extensively researched and documented rural South African town with recognised tourism potential, Dwesa was identified as an ideal testing site for the initiative.

The Siyakhula Living Labs Project's (SLLP) primary objective is to "Develop and field-test a simple, cost-effective and robust, integrated e-business/telecommunication platform, for use in marginalized and semi-marginalized South African communities" (Muyingi, 2006). The secondary objective of the project is to "Build technically skilled human resources in the field of e-commerce to support e-commerce activities in marginalized and semi-marginalized communities". In order to achieve this, the institutions have made use of post graduate students from various disciplines to conduct research on various social, economic, cultural, infrastructural, and literacy issues surrounding the implementation. Amongst the many research projects underway, the project coordinators recognised the need for enhanced understanding of the real requirements of the users. This study aims at addressing that need by contributing to the knowledge surrounding requirements elicitation within a rural context.

7.4 Rationale For The Study

The requirements phase of software development is highly recognised as one of the most critical stages in the software development process, the high costs incurred for correcting errors at later stages, have been attributed to an inadequate requirements elicitation process. Requirements elicitation, although well researched and extensively practiced in urban settings, has proven to be one of the most challenging and vital processes in software development. In contexts that are

characterised by high levels of uncertainty, such as is the case with rural ICT projects, the challenges are certainly more adverse.

The digital divide is a worldwide phenomenon that is high on the agenda of international development bodies. As a result, many non profit organisations have been established as a means of addressing the divide, and developing countries have been inundated with ICT Projects. Although the success rate has not yet been reliably quantified (Heeks, 2002), evidence indicates that the majority of ICT projects in developing countries either partially or completely fail (Korac-Boisvert and Kouzmin 1995; James 1997; *The Economist* 2000; World Bank 1993; Oyomno, 1996) in Heeks (2002). Furthermore, rural ICT projects are almost always initiated by outsiders rather than community members, and there is hence an inclination towards reporting on their successes, while neglecting to adequately present their failures (Heeks, 2002). The degree to which ICTs meet the user's needs and contribute to their social and economic upliftment should be carefully considered when evaluating rural ICT projects. An effective requirements elicitation process can be useful in ensuring that the user's needs are reflected in the design of software for rural users.

7.5 Description of the Study

Four software developers of the SLLP were mandated with the task of contributing to the development of an e-commerce platform for the community in Dwesa. They were each allocated the following responsibilities respectively: To develop and field test a prototype of an e-mall from which entrepreneurs in Dwesa can trade their goods; to develop a localised user interface for the e-mall that the users can interact with to update the content of the e-mall; to develop an e-government portal from which the community can access vital government services via the internet; and finally to integrate all the components of the e-commerce platform into a user friendly web based application.

This research evaluates the process of requirements elicitation within the implementation of the e-commerce platform. A literature review of several requirements elicitation techniques was conducted and thereafter the application of the techniques was reviewed. Based on existing literature on requirements elicitation, and the actual experience eliciting requirements in the rural

scenario, a model was introduced to address some of the challenges of requirements elicitation in rural areas.

This chapter is presented as a case study reflecting the various requirements elicitation techniques applied in the development of the E-Commerce platform. Interviews were conducted with community members and their effectiveness as requirements elicitation techniques was evaluated. The researcher then proceeded to gain an understanding of the approaches undertaken by the software developers of the SLLP in determining the user's needs. The requirements elicitation activities of one of the software developers were directly observed, and thereafter a survey was distributed to the three remaining software developers to evaluate the techniques that they employed to determine the requirements of their respective user's. The result of the survey, as well as documentary evidence from their respective research projects was consolidated to evaluate the reasons for the strengths and shortcomings of the overall requirements elicitation processes. Finally, opinions of project members as they were reflected during correspondence regarding the project were captured as a means of evaluating the project outcomes from different perspectives.

The previous chapter proposed a model for requirements elicitation that categorised the different perspectives of the user requirements into two classes: the users' perspective and the analyst's perspective. The model further distinguished the requirements obtained through the requirements elicitation process into two categories: conceptual and technical. The model then divided the requirements elicitation process into five classes of activities: analysis, specification, design, implementation and operation, and finally, the model associated categories of requirements elicitation techniques to the various combinations of categories. This chapter aims at assessing the requirements elicitation activities of the SLLP for the purpose of verifying the validity of the proposed model.

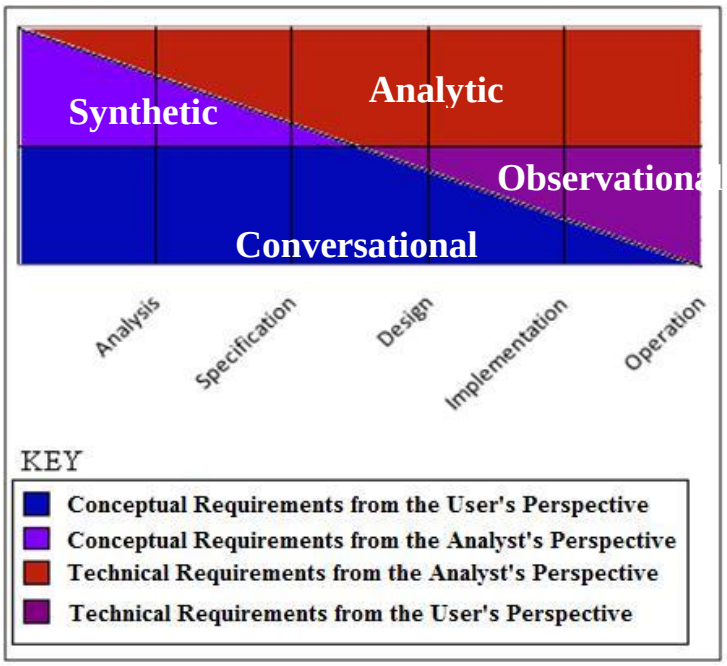


Figure 15: Author's proposed model of requirements elicitation

7.6 Evaluation of Techniques Applied by Software Developers

A survey was distributed to three of the software developers to determine the techniques they applied in defining the user requirements for their respective projects. The fourth software developer's requirements elicitation activities were directly observed and incorporated into the analysis of the survey. As they did not all apply the same software development methodologies, the stages of software development in which the requirements elicitation techniques were applied were determined according to the nature of requirements sought through each technique. Requirements elicitation activities aimed at determining the user profile, understanding the variables that could influence the projects or determine the needs of the users were therefore classified under the analysis phase, while techniques to validate and express the analyst's interpretation of the requirements obtained during analysis were classified under the specifications phase. Techniques applied to ensure that the software design was consistent with the needs and expectations of the users were classified under the design phase, and techniques applied to identify requirements that had been misrepresented or poorly executed in the system's design were classified under the implementation phase. Finally, techniques used to determine the degree of user satisfaction, sustained use, overall goal achievement, and/or solution of the initial problem/need were classified under the operation phase.

7.7 Techniques Applied During The Analysis Phase

During the analysis phase, the most widely used elicitation techniques were interviewing and prototyping. All (100%) of the four developers made use of interviews to gain what 50% of them classified as mostly conceptual requirements and the other 50% classified as equally conceptual and technical requirements. Prototyping was also applied by all (100%) of the developers. Of the four developers that made use of prototypes to elicit requirements during the analysis phase, two of them (50%) classified their prototyping exercises as low fidelity, of which one obtained mostly conceptual requirements, and the other obtained equally conceptual and technical requirements. High fidelity prototyping was applied by only one (25%) of the developers, who classified the requirements obtained through the technique as completely technical. One developer (25%) applied a general prototyping approach that they neither classified as high fidelity nor low fidelity, and described the requirements elicited as mostly technical.

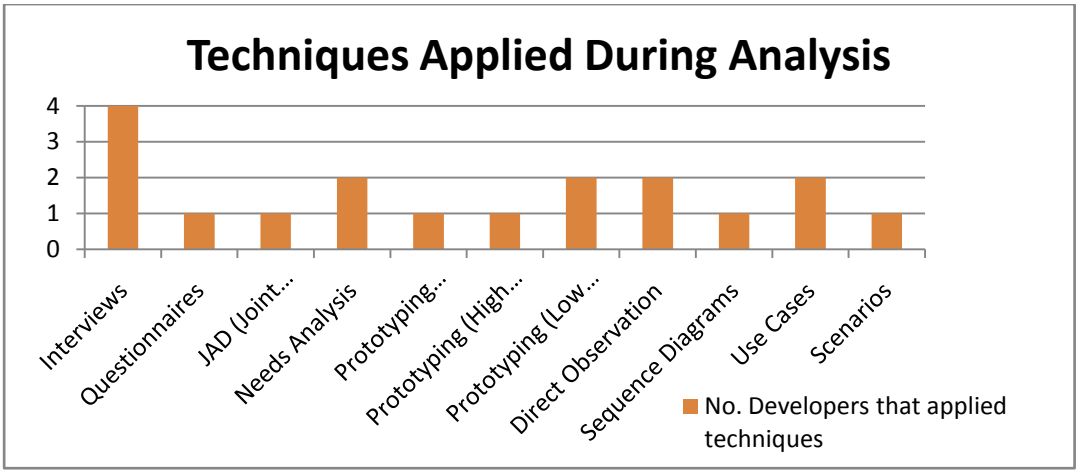


Figure 16: Analysis of techniques applied during analysis phase

Direct observation, needs analysis and use cases followed prototyping and interviews in terms of their frequency of use during the analysis phase, as they were each applied by 50% of the developers. The requirements elicited through these techniques were classified by the developers as equally conceptual and technical. Questionnaires, Joint Application Design (JAD), sequence diagrams and scenarios were the least applied techniques during the analysis phase, with only one developer (25%) making use of each technique during the analysis phase. The requirements elicited

through JAD sessions were classified by the developer as completely conceptual, while the scenarios were classified as mostly conceptual. The requirements elicited through questionnaires were classified as mostly technical, while sequence diagrams yielded requirements which the developer classified as equally conceptual and technical. Figure 16 graphically summarises the frequency of application of techniques by the software developers during the analysis phase.

7.7.1 Summary of the Developers' Classification of the Requirements Elicited During the Analysis Phase

The developers classified the majority (46%) of the requirements as equally conceptual and technical. The number of requirements classified as completely conceptual was the same as those classified as completely technical. However, the quantity of requirements that were classified as mostly conceptual (27%) significantly exceeded those that were classified as mostly technical (13%). A weighting of 1.5 was allocated to completely conceptual or completely technical requirements, and a weighting of 1 to mostly conceptual and mostly technical requirements, as well as a weighting of zero to requirements that are equally conceptual and technical. The survey found that in the analysis phase, 64% of the requirements were conceptual, while 36% were technical, and the ratio of conceptual requirements in relation to technical requirements was therefore 1.78. This weighting was applied throughout the analysis to ensure the consistency of the findings. Figure 17 graphically summarises the developer's classifications of the requirements elicited during analysis.

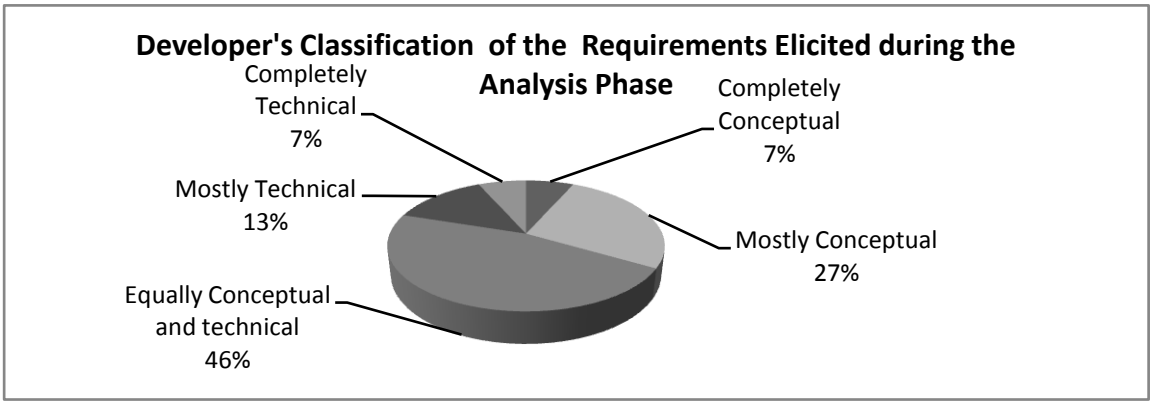


Figure 17: Summary of developer's classification of techniques during analysis phase

7.8 Techniques Applied During the Specification Phase

During the specification phase, the most frequently used technique was prototyping. Three of the four developers (75%) made use of prototyping to communicate and validate their perspective of the proposed systems to the users. During this stage, one of the four developers (25%) classified the requirements elicited through prototyping as equally conceptual and technical, while two of them classified the requirements as mostly technical. During the analysis phase prototyping had obtained requirements which were classified by one of the developers as mostly conceptual. This showed that although the same technique was applied during analysis and specification, the nature of the requirements sought shifted from the technical side towards the conceptual side.

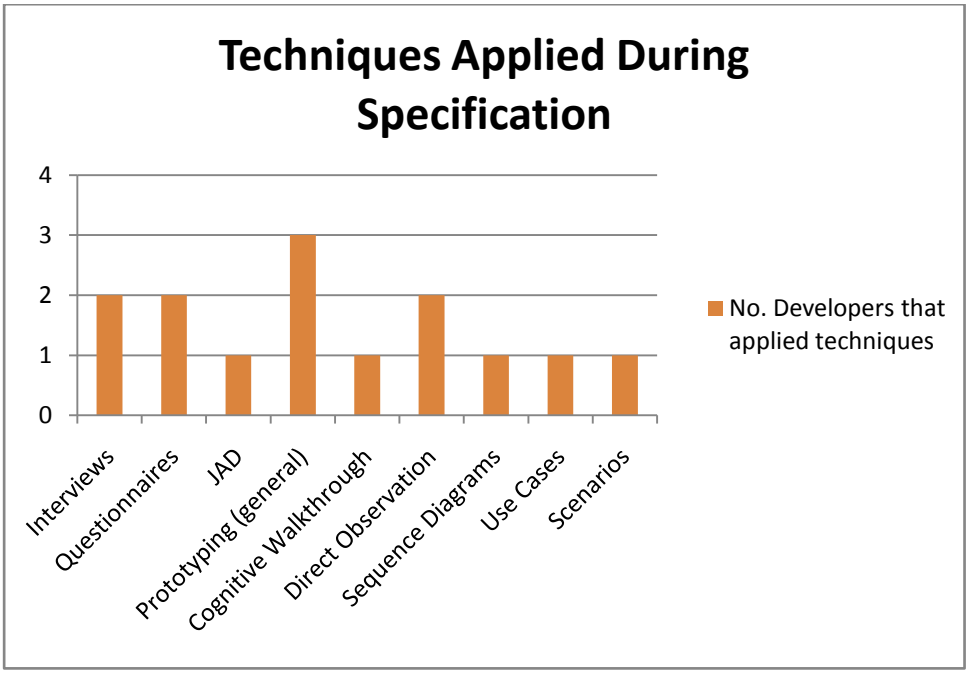


Figure 18: Analysis of techniques applied during specification phase

Interviews, questionnaires and direct observation followed prototyping in terms of the frequency of which they were applied during the specification phase. Each of these techniques were utilised by 50% of the developers. The requirements elicited through interviews and direct observation were classified as equally conceptual and technical by the developers that applied them, as opposed to the analysis phase where they had been classified as mostly conceptual, by one developer, and equally conceptual and technical by the other. The classification for the requirements gained through direct observation remained unchanged, and the requirements derived through questionnaires shifted

contrastingly from mostly technical, to equally conceptual and technical by one developer, and mostly conceptual by the other. Finally, use cases, cognitive walkthroughs, sequence diagrams, and scenarios were each applied only once each during the specifications phase. Use cases and scenarios, which were respectively classified as completely conceptual, and equally conceptual and technical during the analysis phase, remained under the same classification during the specification phase. Figure 18 represents a graphical comparison of the frequency in which the requirements elicitation techniques were applied during the specification phase.

7.8.1 Summary of the Developers' Classification of the Requirements Elicited During the Specification Phase

The developers classified the majority (57%) of the requirements as equally conceptual and technical. There were no requirements classified as completely conceptual, while those classified as completely technical increased to 22% in the specifications phase, in contrast to their proportion of 13% during the analysis phase. In contrast to the analysis phase where the requirements classified as mostly conceptual considerably exceeded those classified as mostly technical, the quantity of requirements in the specifications phase that were classified as mostly technical (22%) significantly exceeded those that were classified as mostly conceptual (14%). During the specification phase only 31.5% of the requirements elicited were categorised by the developers as conceptual, while 68.5% were categorised as technical. The ratio of conceptual requirements in relation to technical requirements during the specifications phase (0.46) was excessively lower than in the analysis phase (1.78). Although many of the techniques applied during the analysis phase were also applied during the specifications phase, the nature of information sought during the specifications phase was more technical and less conceptual than the analysis phase.

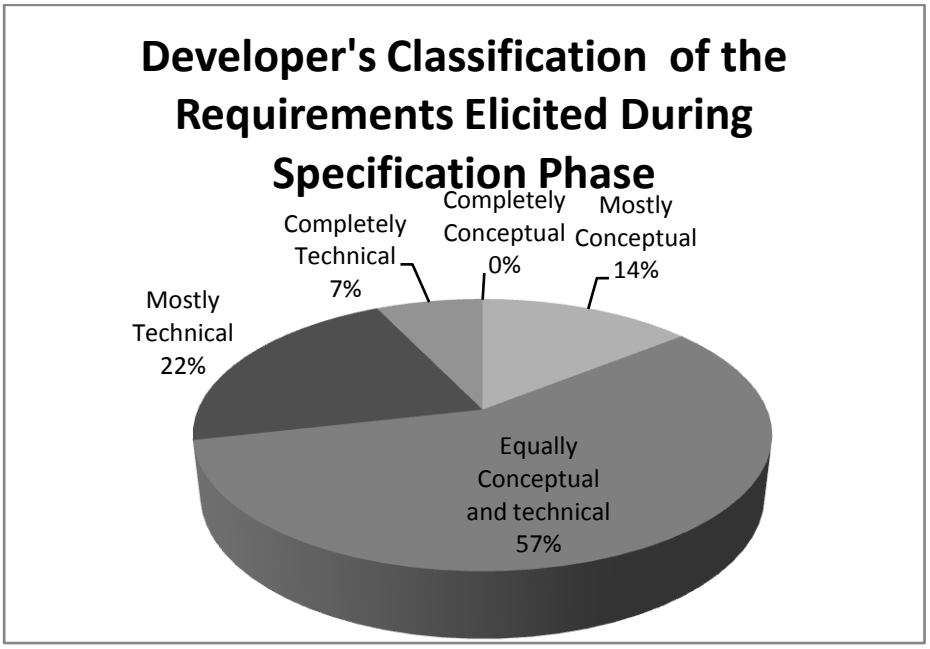


Figure 19: Summary of developer's classification of techniques applied during specification phase

7.9 Techniques Applied During The Design Phase

During the design phase, interviews, prototyping, direct observation, sequence diagrams and scenarios were each applied by two of the four developers (50%). Interviews were classified by one of the developers as mostly conceptual and by the other as equally conceptual and technical. Prototyping was classified as mostly technical. Scenarios were classified as mostly conceptual by both developers that applied the technique. Use cases, acceptance tests and questionnaires were each only applied amongst all the developers. Use cases and questionnaires were classified as equally conceptual and technical, while acceptance tests were classified as mostly technical. Figure 20 provides a summary of the software developer's classification of the requirements elicitation techniques applied during the design phase.

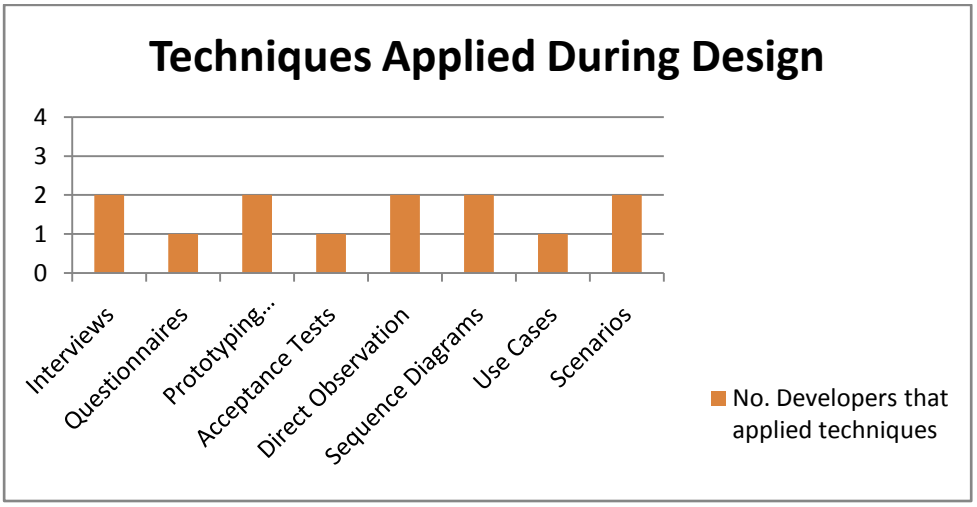


Figure 20: Analysis of techniques applied during the design phase

7.9.1 Summary of the Developers' Classification of the Requirements Elicited During the Design Phase

The developers classified 46% of the requirements they elicited during the design phase as equally conceptual and technical. The requirements classified as mostly technical (31%), exceeded those classified as mostly conceptual (23%) by 8%. The proportion of conceptual requirements in comparison to technical requirements was 0.74. The design phase saw a diversion in the trends of the analysis and specification phase as the requirements elicitation techniques applied yielded more conceptual and fewer technical requirements than the preceding stage. However, this could have been attributed to the fact that the differences in proportions of conceptual requirements in relation to technical requirements obtained between the analysis phase (1.78) and the specification phase (0.46) was exceedingly high. The design phase exhibited a less divergent proportion (0.74) of conceptual requirements in relation to technical requirements when compared to the analysis phase. There was nevertheless a decline in the overall quantity of conceptual requirements obtained and an increase in the technical requirements during the design phase, in comparison to the analysis phase. Figure 21 summarises the software developers classifications of the techniques applied during the design phase.

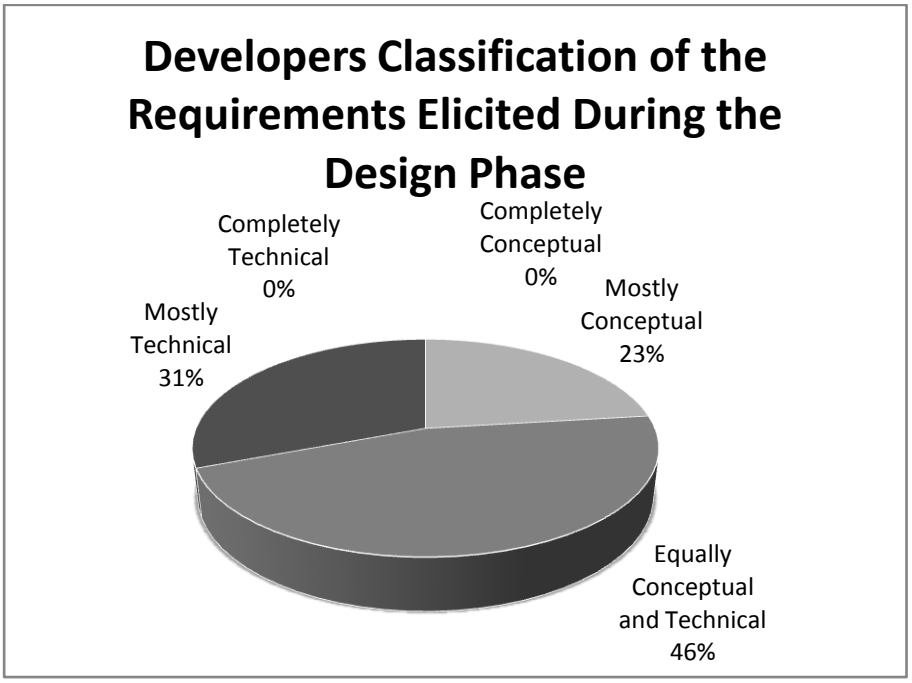


Figure 21: Summary of developer's classification of techniques applied during the design phase

7.10 Techniques Applied During The Implementation Phase

Direct observation, use cases and interviews were the most frequently applied techniques during the implementation phase, and were each applied by three of the four (75%) software developers. All three developers that made use of direct observation and interviews classified the requirements obtained as equally conceptual and technical. The requirements derived through interviews were classified by two of the developers as equally conceptual and technical, and by the third developer as mostly conceptual. Questionnaires, needs analyses and cognitive walkthroughs were applied by two of the four developers (50%). The requirements gained through needs analyses and questionnaires were classified as equally conceptual and technical by all the developers that applied them, while the requirements elicited through cognitive walkthroughs were classified as completely technical by both developers that applied the technique. A summary of the techniques applied during the implementation is provided in figure 22.

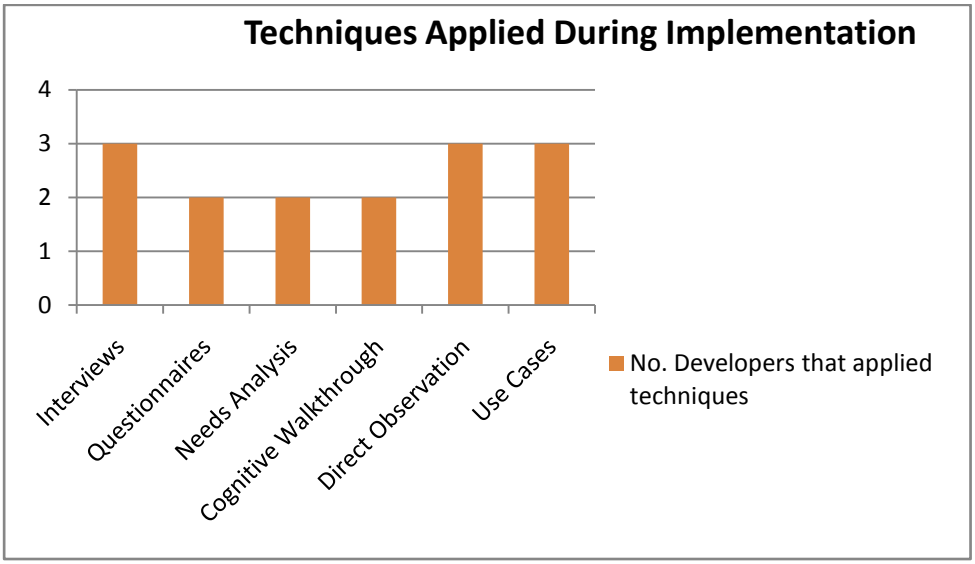


Figure 22: Analysis of techniques applied during the implementation phase

7.10.1 Summary of the Developers' Classification of the Requirements Elicited During the Implementation Phase

The majority (80%) of requirements elicitation techniques yielded requirements which the developers classified as equally conceptual and technical. 7% of the requirements elicited were classified as mostly conceptual, while 13% were classified as completely technical. Overall, the proportion of conceptual requirements in relation to technical requirements was (0.36). This was reasonably lower than the ratio between the technical and conceptual requirements obtained in the design phase. A summary of the software developer's classifications of the requirements elicitation techniques applied during the implementation phase is provided in figure 23.

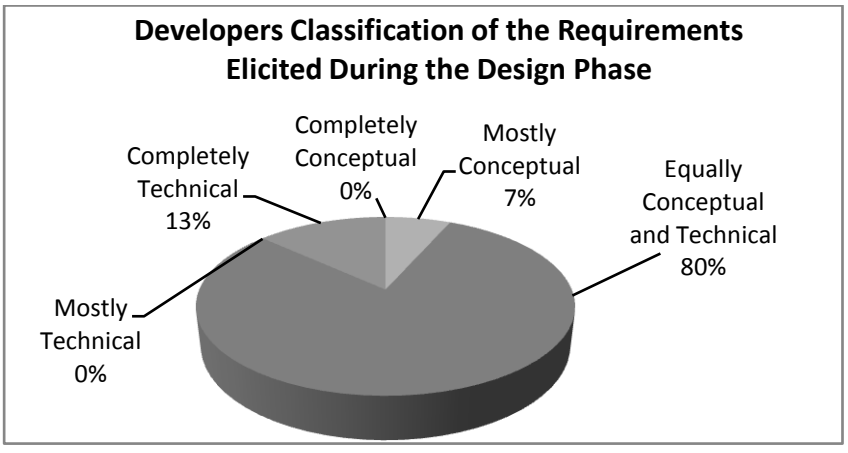


Figure 23: Summary of developer's classification of techniques applied during the implementation phase

7.11 Techniques Applied During The Operation Phase

During the operation phase the developers made use of interviews, questionnaires, prototyping, acceptance tests, direct observation and use cases to determine the degree to which they had achieved their goals and satisfied the needs of the users. Direct observation and questionnaires were the most frequently applied techniques and were applied by three of the four software developers (75%). The requirements gained through the use of questionnaires were classified as completely conceptual by one developer, and equally conceptual and technical by two developers. Direct observation on the other hand yielded requirements that were classified as equally conceptual and technical by two developers, and mostly technical by one developer. Interviews and prototypes were each applied by two of the four developers (50%). Interviews yielded requirements that were classified by one of the developers as mostly conceptual, and by another as equally conceptual and technical. The requirements derived through prototyping activities were classified as mostly technical by one developer and completely technical by the other. Finally, use cases and acceptance tests were each applied only once during this phase and yielded requirements which the developers classified as mostly technical.

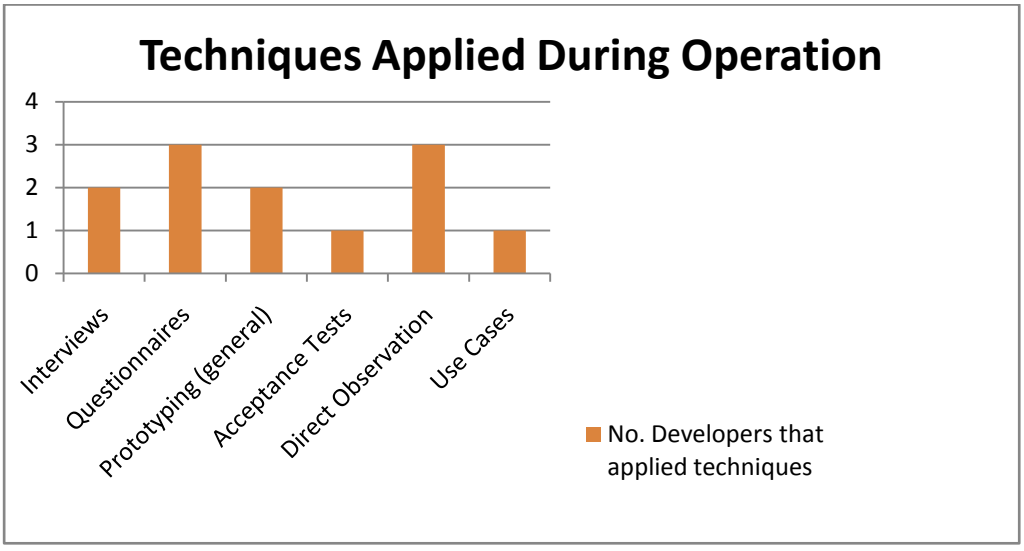


Figure 24: Analysis of techniques applied during the operation phase

7.11.1 Summary of the Developers' Classification of the Requirements Elicited During the Implementation Phase

The majority of the techniques derived requirements that were classified by the developers as equally conceptual and technical (42%). The requirements under the classifications of mostly conceptual, completely conceptual and completely technical each took up 8% of the total classifications. Finally, the remaining 34% of the requirements obtained were classified as mostly technical. Overall, the amount of technical requirements obtained during the operation phase exceeded the conceptual requirements by 26%.

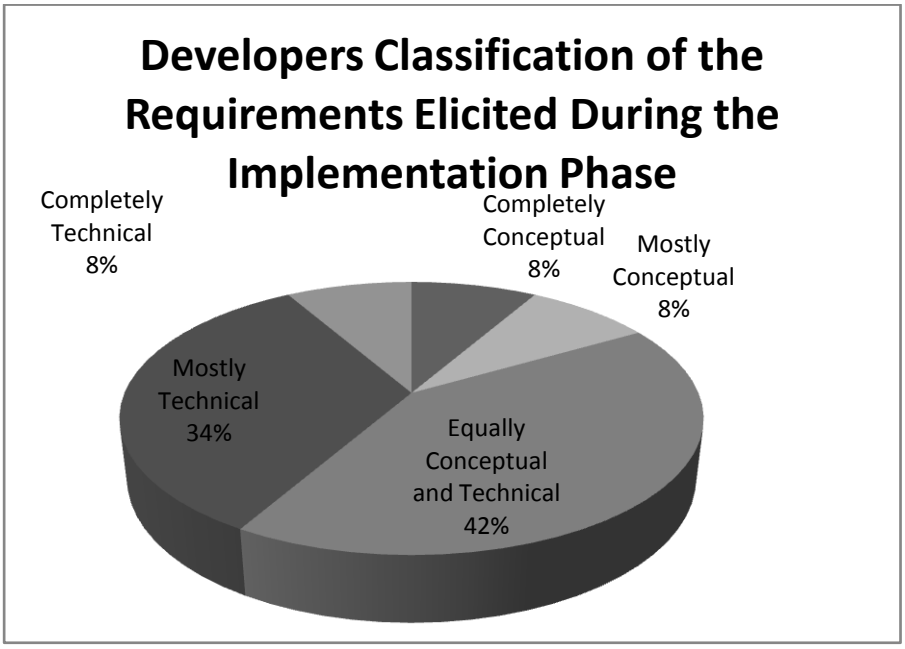


Figure 25: Summary of developer's classification of techniques applied during the operation phase

7.12 Summary of the Entire Requirements Elicitation Process

7.12.1 Summary of Techniques Applied

Interviews, direct observation and questionnaires were the most frequently applied techniques and were utilised in all the stages of the Software Development Lifecycle. Use cases and prototyping followed in terms of the frequency of which they were applied, but were only applied in four of the five stages of the development lifecycle. Amongst the five most frequently applied techniques, all the four categories of elicitation techniques (conversational, synthetic, analytic and observational)

were exercised. Figure 26 summarises the frequency of the application of the various requirements elicitation techniques through the various stages of the Software Development Lifecycle.

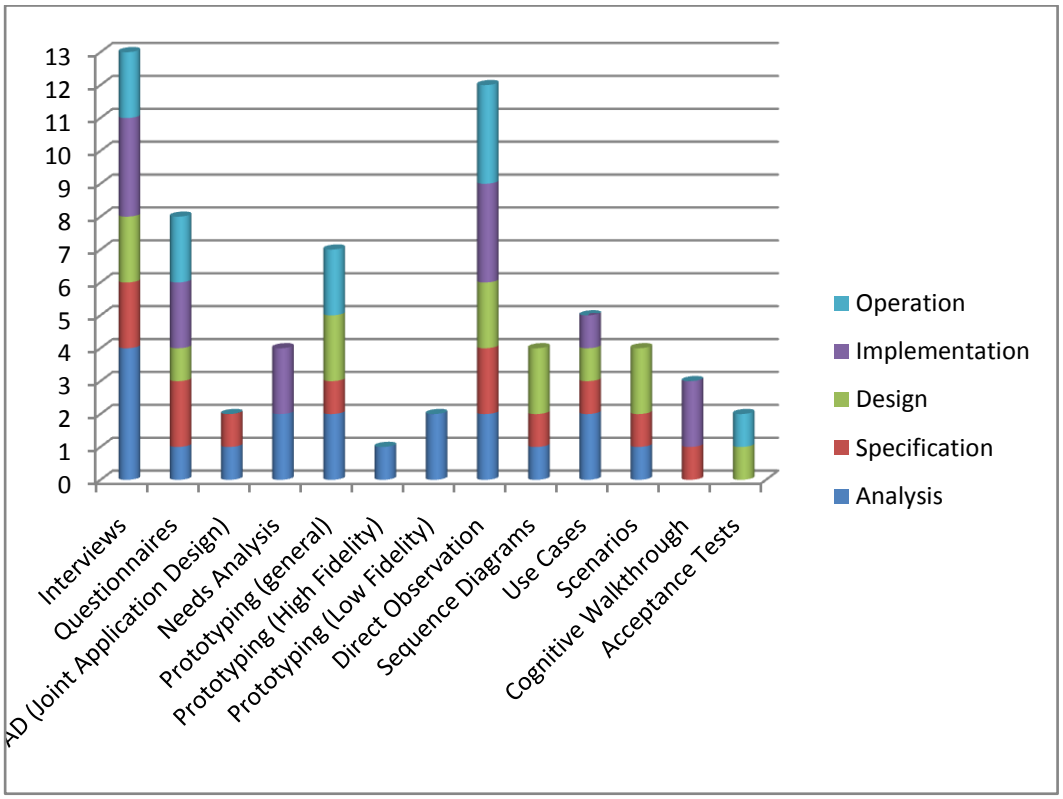


Figure 26: Analysis of techniques applied throughout the development process

7.12.2 Relationship between Conceptual and Technical Requirements

Between the analysis and operation phases, the quantity of requirements classified by the developers as conceptual declined while the quantity of requirements classified as technical increased. The previous chapter presented a model that was based on a theory that knowledge and information is required by software developers from the users throughout all the stages of software development. During these stages various techniques are employed to gain information or knowledge to serve various goals. The theory suggests that as the developers progress through the various stages of software development, there will be a decline in the quantity of conceptual requirements sought, and an increase in the quantity in technical requirements elicited. The results of the survey corresponded with the theory stipulated by the proposed model as it demonstrated the inverse relationship between conceptual and technical requirements through the progression of the software development

lifecycle, while figures 27 and 28 depict the relationship between conceptual and technical requirements through the progress of the lifecycle.

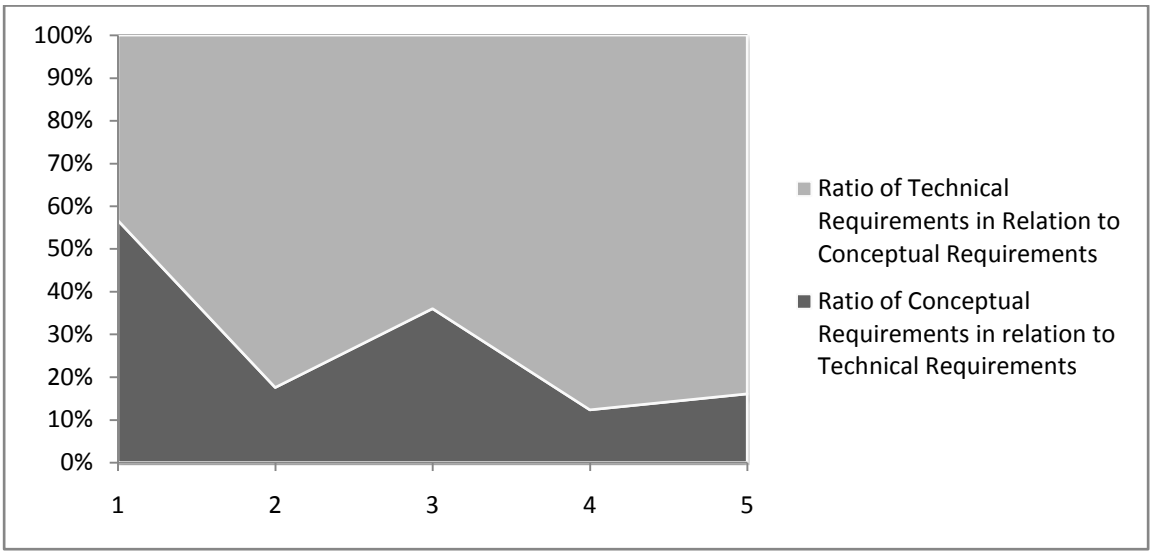


Figure 27: Illustration of the actual proportions of conceptual and technical requirements obtained during the entire project

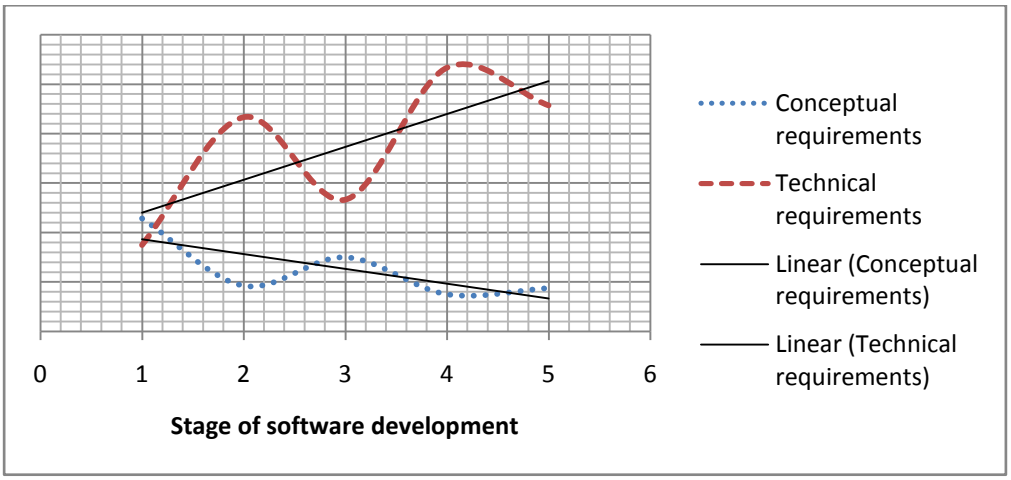


Figure 28: Relationship between conceptual and technical requirements obtained throughout project

7.13 Conversational Techniques: Experience Conducting Interviews in Dwesa

7.13.1 Group Interviews

A field trip to Dwesa was undertaken in May 2006 for the duration of one week. The trip served two aims; firstly to determine the nature of entrepreneurial activities in the area, and explore ways in which ICTs can support rural entrepreneurship, and secondly to assess and document the interviewing process in a rural context against existing requirements elicitation literature. With the assistance of Mrs Gxarisa and Mr Jabe, the project champions at the school where the project is being piloted and Mrs Sothyantya a teacher at the school, three ladies involved in arts and crafts in Dwesa were identified and informally interviewed. The ladies were interviewed as a group. As they were not conversant in English, a group member who speaks the same language as them (isiXhosa) assisted with the translations. The translator was an Honours research student from the Fort Hare University Faculty of Arts. The translator's research was based on the communication issues surrounding rural ICT implementations and the interviews were therefore also of interest to her. Her contribution was therefore not restricted to translation alone, but also included enquiry. The interviewing process was observed passively by a Masters Student from Rhodes University, who was conducting research on project management models within the context of rural ICT projects. Mr Jabe, one of the local project champions, also participated in the interview by asking questions to the ladies that he felt were of relevance. As very little was known regarding the entrepreneurial activities in the area, the interview was semi structured with open ended questions.

i. Knowledge Elicited Through the Interviews

During the interview, the translator probed as to why the ladies had not engaged in the computer literacy training held regularly at Mpume Primary-Junior Secondary School. To this the ladies explained that firstly, due to their age, their vision was poor and they therefore struggled to use the computers. Secondly, they thought the lessons were only targeted at young people. The translator thereafter explained to the ladies how using computers could be of benefit to them. This proved to be quite fruitful because later that afternoon, the ladies all arrived for computer training.

Figure 29: Primary school at Mpume where project is being hosted

Mr Jabe asked many questions that shed a lot of light as to the situation with arts and crafts in Dwesa. He enquired, for example, as to why they had not made use of the crafts centres that were set up in Dwesa, to which they replied that they are not used to working in groups as their crafts are done in their own time, which does not always coincide with one another's timetables. The interviews revealed that the three ladies, Ms Nojongile Ndlumbini, Ms Nowinile Mgidlana and Ms Nophikile Mafenuka all learnt the crafts from their mothers. When asked what they have achieved through the skills, they all mentioned that they had successfully educated their children from income from the crafts. Ms. Mgidlana mentioned that she educated all 5 of her children this way. The ladies said that their children were also involved in making the products because they had seen the contribution that the trade had made to the household. Mr Jabe also added that one of Ms. Mgidlana's children had obtained a tertiary qualification as a result of her mother's efforts and was currently employed at the education circuit office in Dutywa. The ladies mainly make baskets and mats made from the reeds (Imizi and uduli) that grow along the river. The reeds grow seasonally, and can only be harvested after May. Most of the crafts are sold wholesale to shops or in bulk to individuals, any extra stock is sold individually to interested parties. Ms. Mgidlana and Ms. Ndlumbini sell most of their crafts in Dutywa. They mentioned that the reason they don't sell their goods locally is that materials used are readily available in Dwesa, therefore locals prefer to make them themselves rather than to buy them. The ladies therefore target surrounding areas that do not

have access to the raw materials. They determine the prices independently, but these are sometimes subject to negotiation with the buyers. At the time of the interviews, they stated that they incur a return fare of approximately R50 each time they transport their goods by bus to Dutywa. Although they don't keep any physical records of their business transactions, they maintain that their business is profitable. The ladies do, however, keep contact details of their regular customers, whom they contact when items that have been out of stock due to the availability of raw materials become available. One of the issues that came to light through Mr Jabe's questions was the fact that the ladies employ people to harvest the reeds because the place where the reeds grow is quite a long walk from where the ladies live. Given their age, they prefer to focus on making and selling the items, and "outsourcing" the rest of the work. They mentioned that the difficulty in this is that there is a special skill involved in cutting the reeds, and sometimes they cannot find the necessary skills. They sometimes resort to buying the reeds from local entrepreneurs that sell them for a living. The ladies mentioned that the market fluctuates, the peak seasons are from June to July, then again from November to January. They attribute the fluctuations to the fact that a lot of their goods are used as gifts for ceremonies such as weddings, funerals, initiations etc, and these ceremonies are most commonly held in those months.

By the time the interview had been concluded, more entrepreneurs had arrived and introduced themselves to the group to present their craft work. These crafts included a shirt with beaded collars and cuffs, placemats, a bottle that had been decoratively wrapped in wire patterns, a knobkierie and spear that had been decorated with beadwork, intluza (which is a reed instrument used to strain xhosa beer) and thick reed mats that are used for chopping vegetables. The entrepreneurs were invited to attend a group discussion the following day to discuss participation in the project. The meeting aimed at establishing whether or not they were willing to participate in the project, and how they would cooperate with one another to make the venture a success. One of the goals of the meeting was also to demonstrate how online businesses operate. The research group browsed several online stores to demonstrate how the entrepreneurs could develop their business through the E-Commerce platform. The group of entrepreneurs concluded that they were willing to work together, and nominated Mrs Sothyantya, a teacher at the school and fellow entrepreneur to be their representative.

Figure 30: The researcher, translator and two of the interviewees.

Much time was spent training teachers and community members, and identifying more crafters to participate in the project. The group was quite pleased to note that Ms. Mgidlana, one of the ladies that had been interviewed the previous day, had returned to attend computer literacy training. She was quite ill but had insisted on at least making a presence and excusing herself for the following day's training, as she intended to go to the clinic for treatment. During the visit, some of the project group members identified a group of ladies actively involved in their craftwork. The ladies were approached and introduced to the project, and the potential ways in which E-Commerce could benefit their businesses was explained to them. They expressed a keen interested in getting involved, and enquired as to when the project would begin. They were informed that the purpose of the trip was to identify entrepreneurs, and the next step would be to plan the project based on the entrepreneurs that had identified. They were also told that the teachers at Mpume Junior Primary-Secondary School would communicate the date of the group's return and any other relevant information.

Figure 31: Interview conducted with crafters at work in Dwesa

7.13.2 Individual Interviews

It had been noted that most of the entrepreneurs identified had been elderly women. The researcher therefore enquired as to whether there were any men in the area that were involved in similar activities. The ladies directed the group to a gentleman in a part of Dwesa called Ngoma, Mr Mpatyu Nkohla, who carves canes (similar in appearance to walking sticks) that are commonly used for religious and traditional ceremonies. Mr Nkohla is well known in the area, and it didn't take long to find him. The group was well received by Mr Nkohla who showed them a broad variety of his handwork. He was notified of the entrepreneurship project, and he expressed a strong interest in participating. It was established that Mr Nkohla had quite a broad client base, and he received orders from as far away as Cape Town. He expressed his desire to work with people in improving his products, and also mentioned his desire to train people in his skill. Unfortunately, he also explained that he had failed to find people willing to work with him.

Amongst the issues brought to light in the interview with Mr Nkohla, one of the most notable was the fact that he struggles to obtain wood to carve his goods because of the conservation restrictions placed on the resources. He mentioned that the best quality wood is in the nature reserve. Initially they were required to pay for permission to obtain wood from the reserve. Mr Nkohla did not mind paying for access to the wood in the reserve, but even that prospect had recently been revoked by the

management of the nature reserve. He currently has to walk long distances to get wood of an inferior quality. He also mentioned that an opportunity had existed a few years ago for him to purchase equipment to improve his work, but he couldn't afford it at the time. (The equipment cost approximately R1000). He was also asked whether he kept any financial records for his business, to which he answered that he did not keep any records, nor did he have a bank account. He explained that most of the money that he earns is used immediately. He also added that because he was not in the habit of saving, he had never seen the need to possess a bank account. He did however mention that since his children live far away, and would like to send him money, he would consider getting a bank account.

7.13.3 Evaluation of Conversational Requirements Elicitation Techniques: Individual and Group Interviews

i. Extensive Depth and Breadth

The interviews provided extensive breadth and depth into understanding the conditions, challenges and opportunities within the community that could influence the ability of the E-Commerce platform to support their entrepreneurial activities. The researchers were able to identify a broad variety of entrepreneurial activities taking place in the area, determine the general manner in which business is conducted amongst community members, and the community's challenges in accessing resources as well as distributing their products.

The interviews helped to give direction to the research. The information derived from the interviews emphasized the fact that the success of the project requires a lot more than just a technically sound e-commerce platform. The interviews revealed that the development of the E-Commerce project would most likely require basic guidance or training in business management as the activities that are currently on a very small scale may expand considerably. Interaction with the interviewees highlighted a need to examine matters pertaining to logistics, marketing and quality control, as well as the need to investigate the purchasing of equipment to make the work of the entrepreneurs easier. As the time taken to produce a single item is quite lengthy the entrepreneurs need to employ more cost effective means of production in order to price their products competitively.

ii. Face to face communication

The face to face contact with the community members was a strong characteristic of the interviews. It allowed the researcher to sense emotions such as enthusiasm or reluctance, which may not have been articulated in non verbal techniques such as questionnaires. Although almost all the entrepreneurs were elderly, they all demonstrated an energy and eagerness to participate in the project and seemed to be open to change and eager to learn.

iii. Translation Challenges

As the interviews went through a translation process, it was observed that a lot of meaning was lost or miscommunicated. For instance in some cases the interviewee's long and detailed response was translated to the interviewer in a brief summarised format. Although the message was conveyed, the ability to critically analyse and document the response effectively was compromised by the language barrier.

During the interview, it was also observed that the translator sometimes used strong emotive language that led the interviewees to believe that e-commerce was a certain solution to their socio economic challenges. This was discussed after the interviews had been concluded and it was agreed that it would be reinforced that although the project had the potential to improve their circumstances, success was not guaranteed, and much effort was necessary from the researchers as well as the community to obtain the desired results.

Challenges resulting from the translation process drew the author to the literature of George, Hoffer and Valacich (1998), who emphasise pre-interview planning. They state that the goals and outcomes should be well communicated to all participants prior to the interview. The challenges raised through the translation process could have been avoided if the author had discussed the research principles and objectives with the translator prior to the interviews. This would have ensured that the principles and objectives regarding the research were aligned, and hence the likelihood of false expectations amongst the community members would have been minimised.

iv. Group Dynamics

Group sessions were successful in promoting unity and cooperation amongst participants. Individuals that exhibited strong leadership qualities through the amount of positive influence they had on the group were identified as potential project champions. The group sessions demonstrated the ease in which the community could self organise to achieve an objective. Ms. Mgidlana was identified as a potential project champion as she was well acquainted with most of the entrepreneurs in the group, and she expressed a great deal of enthusiasm in the project. Mrs. Sothyantya, who had strong links with the community as a teacher and entrepreneur, was nominated by the group to represent their interests in the project.

7.13.4 Ability of Conversational techniques to elaborate the conceptual layer based on the user's perspective

Conversational techniques proved to be very effective in elaborating the conceptual issues concerning the project. Conversation provided a very natural means of interacting with the community and engendered trust and unity amongst stakeholders. The interviews provided a thorough understanding of the context of the implementation, and were well suited for the analysis stage of development as the knowledge they provided had the ability to form the foundation for subsequent software development activities. The use of conversational techniques during analysis also ensured that the prospective user's needs were expressed by the users as opposed to being assumed by the analysts. As one of the challenges in rural software implementations is that the technology tends to be "pushed" on to rural communities with little regard of their real needs. Engaging extensively with the community to understand their needs is likely to encourage ownership of the outcomes by the user community, especially given the fact that they were consulted from the onset. Although the techniques applied did have their shortcomings, these could be prevented or overcome through more effective planning and implementation. The study therefore concludes that conversational techniques support requirements elicitation by allowing analysts to gain extensive knowledge of the users, their activities and the environment within which they operate. This conceptual knowledge is best elicited during early analysis stages of software development as it informs the later software development activities.

7.14 Evaluation of Synthetic and Observational Requirements Elicitation Techniques

A prototype of the proposed system was demonstrated to the entrepreneurs in Dwesa in November 2006. The exercise allowed the author to evaluate prototyping as a synthetic requirements elicitation technique, as well as to make comments on observational techniques through the experience. A literature review conducted prior to the prototype demonstration had revealed that the disadvantage in constructing a high fidelity prototype was the extensive amount of time involved in developing the prototype, as well as the attachment and resistance to criticism of the developers, associated with the development timeframe. Furthermore, the literature stated that analysts should encourage negative feedback from their prospective users in order to identify as many potential flaws in the system as possible.

7.14.1 High Fidelity Prototyping and Direct Observation

During the prototype demonstration, the author observed that the audience were generally impressed with the prototype, and offered no criticism. Although the author assumed that the reasons for the failure to criticise the prototype were founded on the inexperience and lack of exposure to similar technologies, this may have been further heightened by the fact that the developer had placed a lot of time and effort into the development of the prototype, and may have been less open to criticism. According to Zhang (2007), synthetic elicitation techniques, such as prototyping, aim at allowing analysts to validate their view of the users' problem space with the users themselves by combining conversation, observation, and analysis into single techniques that analysts and stakeholders can apply to communicate and interact amongst each other to reach a common understanding. The inability of the exercise to evoke the opinions of the users compromised the ability of the prototyping exercise to achieve this purpose. As the knowledge obtained through observational techniques is highly subjective, the effective application of synthetic techniques is critical to ensure that the user's views are reflected in the system design. Other synthetic elicitation techniques such as use cases and scenarios were applied by some of the developers in the absence of the prospective users. As a result the user's opinion was not well represented in the overall design of the system.

7.15 Evaluation of Analytic Requirements Elicitation Techniques:

The software developers made use of traditional analytic techniques such as use cases, sequence diagrams and scenarios to gain technical requirements based on their academic and professional knowledge of software development. As the application of these techniques was informed by the software developer's knowledge in the field, as well as on the requirements obtained through prior interaction with the users, the techniques in many cases were less interactive and conducted mainly in the absence of the users. The author was therefore unable to directly observe and comment on the implementation of the techniques. However, as analytic techniques engage experts' knowledge or the information surrounding the proposed system, this study also evaluates the discussion forums that the project members used to provide feedback and make recommendations for the improvement of the project activities under the category of analytic techniques.

An e-mail list that includes the contacts of all the stakeholders of the SLLP provides a means of sharing knowledge or information, exploring problems and solutions, and updating group members on the progress of the project. Occasionally, issues pertaining to the project are discussed extensively amongst the project group members. The e-mail list provides a platform upon which group members can generate and debate ideas based on the experience and knowledge that they have gained through their respective research projects. One such issue was raised towards the end of 2008 regarding the need for increased interaction between the software developers and the prospective users. The discussion was raised by one of the researchers on the project who had held focus groups with a host of community members. Through her interaction with some of Dwesa's community members that were involved in arts and crafts, one of the researchers had discovered that many of the prospective users of the e-mall felt disappointed by the fact that they had not been kept up to date with the progress of the project. They also stated that their expectations had been raised and yet none of the benefits of the project had been realised. This issue had also been communicated by the representative who had been nominated by the crafts group from Mpume Primary-Junior Secondary School. She had stated that although the group was losing confidence in the project, they were still willing to participate. Through the e-mail forum, many of the project group members asserted their opinion on the issue based on the knowledge and experience they had gained through their research. The diverse viewpoints resulted in recommendations that had the potential to enhance ownership of subsequent versions of the software by the prospective users. Although the knowledge gained through analytic techniques was documented in the absence of the users, its

foundation on expert knowledge as well as the knowledge gained through previous interaction with the users made it less subjective than observational techniques.

7.15.1 Assessment of the Requirements Elicitation Process within the Siyakhula Living Labs Project with Respect to the Proposed Model

The software developer, whom the author of this research directly observed, produced a technically sound software application. The actual software developed, as well as the corresponding research report fulfilled the requirements of a Master's degree. Unfortunately the needs of the users were not adequately addressed by the software. Communication with the users during the operation phase of the software development lifecycle revealed that the users did not feel that their expectations of the system had been met. This may have been attributed to the fact that the objectives of the software developers were not well communicated to them and their expectations exceeded the software developer's objectives.

The requirements elicitation processes applied by the software developers of the SLLP were successful in the fact that the technical quality of the software was high, and the developers gained and extensive background on the user's environment. They utilised a broad range of technique from all four classes of elicitation techniques (conversational, synthetic, observational and analytic). The techniques were selected appropriately through the various stages of software development to achieve the goals of the respective stages. Conversational techniques were very effective in determining conceptual requirements based on the users' perspective of the problem domain. Similarly, analytic techniques such as use cases were well suited in defining the technical requirements based on the analysts' perspective and knowledge. However, contrary to the suggestions of the proposed model, analytical techniques were also effective in determining conceptual requirements based on the analysts' perspective. Observational techniques yielded technical requirements based on the analyst's view of the project, while both synthetic techniques elaborated on technical requirements based on the users' perspective.

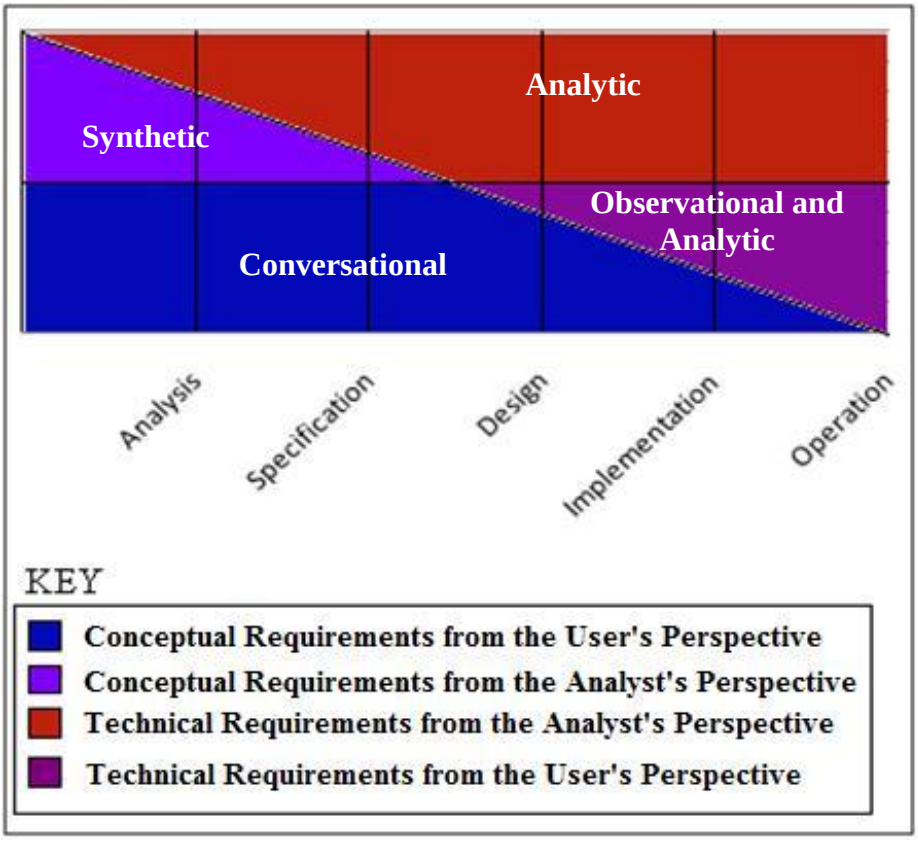


Figure 32: Modified version of the proposed model

The developers' activities corresponded with the major theory presented in the proposed model, and were able to support the model. The only modification to the proposed model arising from the analysis of the SLLP requirements elicitation activities was the inclusion of analytic techniques as a means of obtaining conceptual requirements from the analyst's perspective.

The major weakness in the elicitation activities of the software developers of the SLLP was that they did not sufficiently involve the users in the application of the synthetic techniques. Synthetic techniques serve to merge the analysts view with the users view for a more unified understanding of the proposed system. The users' view contributed mainly during the analysis phase, and the software developers did not make enough effort to validate their view of the system with the users during the subsequent stages. Failure to adequately involve the users through the definition of the system requirements may have been one of the factors that contributed to the high expectations of the users and the inability of the final software to meet their expectations. Synthetic techniques such as scenarios and use cases should have been carried out with the input of the users. The only

synthetic technique that involved the users was prototyping, which failed to yield sufficient feedback from the users.

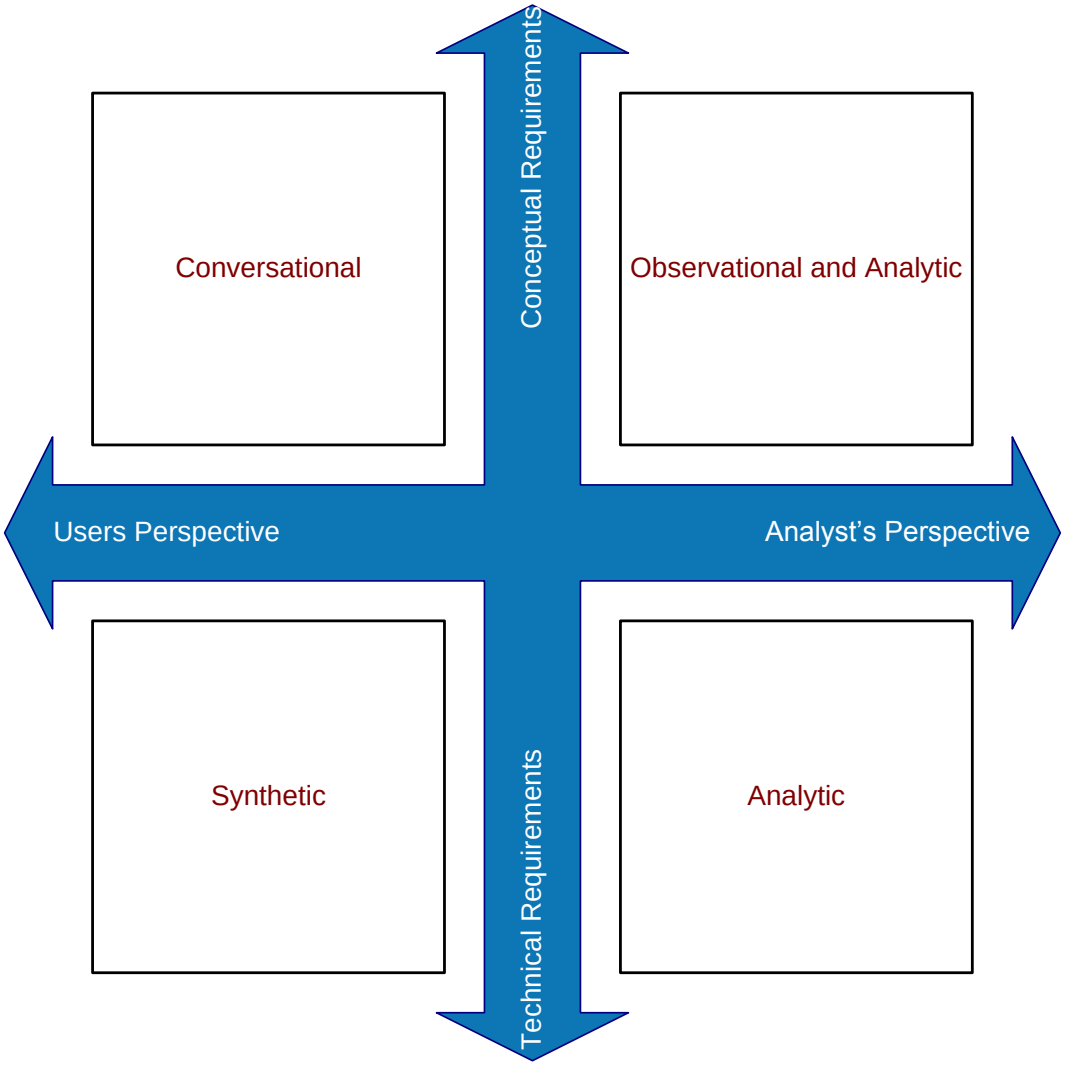


Figure 33: Modifications to requirements perspectives matrix for proposed model

7.16 Conclusion

The review of the requirements elicitation activities of the SLLP's software developers showed that although the software developers did not follow a specific requirements elicitation model, their activities inadvertently corresponded with the proposed model. Furthermore, the author's experience and observations regarding the various categories of techniques (conversational, observational, synthetic and analytic) allowed the author to support their application in the various

quadrants of the model. The study also enabled the modification of the proposed model to accommodate analytic techniques as a means of determining conceptual requirements from the analyst's perspective. Finally the study identified the failure to interact effectively with users in the application of synthetic techniques as the major shortcoming in the requirements elicitation activities of the SLLP, and the underlying cause for the users' dissatisfaction. This shortcoming exacerbated the fact that translation challenges had contributed to the unrealistic expectations of the users when the project was first introduced to them.

Part 3: Conclusions, Recommendations and Directions for Future Research

Chapter 8

Conclusion

8.1 Background

Although the potential for Information and Communication Technologies (ICTs) to empower rural communities and address the problems associated with their marginalisation has been broadly recognised, the benefits are yet to be realised. Over the past decade, rural societies have been flooded with ICT projects that have aimed at addressing the digital divide. Today, many of the communities that have been a testing ground for developmental ICT projects remain impoverished and marginalised, while the organisations that initiated the projects continue to celebrate the accomplishments of their projects. This conflict results from the development of ICTs that achieve the goals of researchers rather than the goals of the users.

An effective user requirements elicitation approach should ensure that the needs of the users are reflected in the design of the software application and hence prevent a situation where the developer's goals are achieved while the user's needs are not. To achieve this, requirements elicitation methodologies must encourage interaction and feedback from the users throughout the software development process. Traditionally, requirements elicitation involves the users early in the software development lifecycle, while developers tend to exclude the users in subsequent stages of the software development lifecycle.

This study has introduced a model for requirements elicitation based on the review of literature surrounding rural ICT implementation as well as requirements elicitation approaches. The contributions of the research are founded on literary evidence from academics in the field, as well as the author's experience observing and participating in the requirements elicitation activities of a rural software development project. Components of this research have also been reviewed by the academic community at a conference, and published amongst the conference proceedings.

8.2 Evaluation of Research Outcomes

Requirements elicitation has been cited as one of the most critical elements of software development. An effective requirements elicitation process ensures that the software is developed

according to the needs of its prospective users. Software that does not meet the needs of its users stands a high likelihood of being abandoned. The main research problem attempted to establish a means of determining user requirements that accommodates users in rural areas. This was based on the premise that the application of existing requirements elicitation techniques and methodologies could not simply be migrated and applied to rural settings without considering the numerous fundamental differences between typical urban users and their rural counterparts.

In an effort to address the main research problem, an extensive review of the literature of experts in the field was conducted. In order to justify the focus of this research, the first goal of the literature review was to distinguish rural ICT implementations from typical urban implementations. The review addressed the first sub problem of the research by substantiating the fact that issues pertaining to education and literacy levels, previous access to similar technologies, language barriers and various socio economic challenges necessitated a deeper review into the adaptation of requirements elicitation techniques and methodologies to suit a rural context.

In response to the second sub problem of the research, the next goal of the literature review was to explore the applicability of a variety of existing requirements elicitation techniques to a rural context. The review provided an extensive comparison between a range of requirements elicitation techniques and established the fact that, although many techniques could be applied to software development projects in rural environments, their effectiveness was influenced by a host of variables. The review concluded that although each individual technique reviewed could be contextualised to a rural setting, reviewing literature surrounding the techniques could not adequately improve the process of requirements elicitation for rural software development.

To address the third and fourth sub problems, the author conducted an in depth review of requirements elicitation methodologies that bore the potential to address the challenges in eliciting user requirements in rural areas. The strengths and weaknesses of each of the methodologies were assessed and thereafter consolidated into a single context independent model for requirements elicitation. Recommendations made by conference reviewers as well as delegates allowed the author to further refine and improve the model. The model was further evaluated and modified in subsequent chapters by making reference to the failures and success in the requirements elicitation activities of the software developers of the Siyakhula Living Labs Project (SLLP).

8.3 Contributions of This Research

The research has contributed to the body of knowledge twofold: firstly, by integrating theories and experiences of researchers in the field of user requirements elicitation into a literature review to accommodate the emerging trend of ICTs for development, and secondly, by presenting a model that can effectively guide the process of requirements elicitation in rural contexts.

Literature in requirements elicitation is generally sparse, and is founded on decades of research that has been largely centred on urban user communities from developed countries. Organisations or individuals attempting to initiate rural software development projects have very little documentary guidance to inform their actions. As a result, a high failure rate for rural ICT projects exists. The situation is perpetuated by the inclination of researchers to report on their success rather than their failures. The literature review provides a compilation of the opinions, theories and experiences surrounding rural ICT implementations, an extensive review of the application of traditional requirements elicitation techniques in rural areas and an in-depth comparison of requirements elicitation methodologies. The availability of such knowledge can assist researchers in their endeavour to implement ICT projects as a means of addressing the challenges associated with marginalised and impoverished communities.

The research also presents a model that encompasses the strengths of the various requirements elicitation methodologies and eliminates their weaknesses based on the limitations of their application in a rural context. The model is founded on the knowledge uncovered through the literature review, and has been validated through the observations made during the interaction with the users and software developers of the SLLP. The model is context independent, and not geared towards any specific environment, and therefore has the potential to support the process of requirements elicitation under a broad range of conditions. One of the key elements that distinguish the model is the manner in which it enforces interaction with the users at all stages of software development. This factor is likely to encourage ownership of ICT applications by their prospective users and may reduce the chance of the users feeling dissatisfied with the end product.

8.4 Directions For Future Research

8.4.1 Independent Testing of the Model

The assessment of the model presented in this research was based on existing theories and methodologies of requirements elicitation, the author's experiences applying various requirements elicitation techniques, as well as the requirements elicitation activities of the four software developers of the SLLP. Although the model was informed by a broad range of sources, the research was nevertheless qualitative and therefore potentially subjective. A need therefore exists for further testing and validation of the model under conditions that will encourage less partial results. Furthermore, the model has been developed to be context independent. This can only be validated by applying the model in contrasting contexts.

8.4.2 Commercial Aspects of E-commerce Applications

Although the arts and crafts of the entrepreneurs have been made available to the public via online stores on the e-mall that was developed by the SLLP, the demand for their goods is so low that most of the entrepreneurs are not even aware that the e-mall exists. Furthermore, one group of entrepreneurs that was fortunate enough to obtain a large order through their online store, had difficulty in meeting the requirements of their client within the required timeframe. As a result, in an effort to preserve the image of the project and the online store, project members from the SLLP collaborated with the group to assist them with the production of the goods. This illustrates the fact that addressing requirements elicitation is only one step in improving the adoption and sustained use of the e-commerce platform. This research's emphasis was predominantly on the development of software that addressed the needs of its users. Although this is an important aspect in encouraging the sustained use of a system, it must be recognised that a system is composed of numerous parts which each require due consideration. Further research is therefore necessary for the examination of factors relating to the development of business management and entrepreneurship skills, product quality assurance and control, the establishment of reliable distribution mechanisms, and the advertising and promotion of the website.

8.5 Summary

ICTs have the potential to improve the livelihoods of marginalised communities, however the benefits can only be realised when they are developed in a manner that appropriately engages with the prospective users from the conception to the completion of the project to understand their needs at each stage of development. An effective requirements elicitation process is one that enforces the balanced interaction between the users and analysts. This will ensure the unity of the users and analysts goals and expectations and will result in an end product that satisfies the objectives of both groups.

References

Alvarez, R., & Urla, J. (2002). Tell Me a Good Story: Using Narrative Analysis to Examine Information Requirements Interviews during ERP Implementation. *The Database for Advances in Information Systems , 33* (1), 38-52.

Atweh, B., Kemmis, S., & Weeks, P. (1998). *Action Research in Practice: Partnerships for Social Justice in Education.* London: Routledge.

Avison, D., & Myers, D. (2005). Qualitative Research. In D. Avison, & J. Pries-Heje, *Research in Information Systems* (pp. 239-253). Amsterdam: Elsevier Butterworth-Heinemann,.

Blignaut, P., Burger, A., & De Wet, L. (2002). Comprehension and Usability Variances among Multicultural web Users in South Africa. *CHI 2002.* Minneapolis: ACM.

Boehm, B. W. (1988). A Spiral Model of Software Development and Enhancement. *IEEE Computer , 21* (5), 61-72.

Boehm, B. W. (1981). *Software Engineering Economies.* New Jersey: Prentice-Hall.

Bourgess-Waldegg, P., & Scrievener, S. A. (1998). Meaning; the central issue in cross cultural HCI design. *Interacting with Computers , 9* (3), 287-309.

Brooks, F. P. (1987). No Silver Bullet: Essence and accidents of Software Engineering. *IEEE Computer , 20* (4), 12-19.

Browne, G. J., & Rogich, M. B. (2001). An empirical Investigation of User Requirements Elicitation: Comparing the effectiveness of Prompting Techniques. *Journal of Management Information Systems , 17*, 223-249.

Burns, R. N., & Dennis, A. R. (1985). Selecting the Appropriate Application Development Methodology. *SIGMIS Database* , 19-23.

Centre for Technology In Government. (1998). *A survey of System development Process Models.* New York: University of Albany.

Christel, M., & Kang, K. (1992). *Issues in requirements Elicitation.* Pittsburgh: Carnehie Mellon, Software Engineering Institute.

Collins, K. (1999). *Participatory Research-A Primer.* South Africa: Prentice Hall.

Collis, J., & Hussey, R. (2003). *Business Research: A practical Guide for Undergraduate and Postgraduate Students* (2nd Edition ed.). New York: Palgrave Macmillan.

Cooper, A. (1997). *Goal Directed Design.* Retrieved December 1, 2008, from CHI-SA Special Interest Group: http://chi-sa.org.za/articles/Goal Directed Design.

Courage, C., & Baxter, K. (2005). *Understanding your users.* Amsterdam: Morgan Kaufmann.

Davies, A. M., & Hickey, A. M. (2004). a Unified Model of Requirements Elicitation. *Journal of Management Information Systems , 20* (4), 65-84.

Davies, A. M., & Hickey, A. M. (2002). Requirements Elicitation and Elicitation Technique Selection: A Model for Two Knowledge-Intensive Software Development Processes. *Proceedings of the 36th Hawaii International Conference on Sysytem Sciences.* IEEE.

Davis, A. M. (1995). *Principles of Software Development.* New York: mcGraw-Hill.

Davis, F. D. (1986). *A Technology Acceptance Model for the Empirically Testing New End User Information Systems.* PHD Thesis, Sloan School of Management, Massachusetts.

Davis, F. D. (1989). Perceived Usefulness, Perceived ease of Use, and End User Acceptance of Information Technology. *MIS Quarterly , 13* (3), 319-339.

Disse, H. (2001). *Dialogical Systems Design Across Cultural Boundaries.* PHD Thesis, University of Hamburg.

Erickson, T. (1996, July). Design as Storytelling. *Interactions of the ACM* , 31-35.

Fals Borda, O., & Rahman, M. A. (1991). *Action and Knowledge, Breaking the Monopoly with Participatory action Research.* New York: Apex Press.

Fay, D., & Palmer, R. (2002). The Residents. In D. Fay, H. Timmermans, & R. Palmer, *From Conflict to Negotiation.* HSRC Press.

Fay, D., Timmermans, H., & Palmer, R. (2002). *From Conflict to Negotiation.* HSRC Press.

George, J. F., Hoffer, J. A., & Valacich, J. S. (1998). *Modern Systems Analysis and Design.* Boston: Addison-Wesley.

Government, T. C. (1998). *A Survey of Systems development Process Models.* University of Albany.

Health Systems Trust. (2005). Retrieved August 10, 2005, from http://www.hst.org.za/publications/535.

Heeks, R. (2002). *Failure, Success and Improvisation of Information Systems Projects in Developing Countries.* Manchester: Manchester University.

Hemson, D., Meyer, M., & Maphunye, K. (2004). *The Provision of basic Infrastructure Services.* Position Paper.

Hugo, J. (2000). *Himan Computer Interaction- A new Challenge for the South African IT Industry.* Retrieved from CHI-SA: http://www.CHI-SA.org.za/articles/HCIChallenge.htm

IEEE. (1990). IEEE Standard Glossary of Engineering Terminology. New York: IEEE.

International Telecommunications Union. (2003). *Internet Indicators: Hosts, Users and Number of PCs.*

ITEPA. (2004). *Recommendations for a Pro Poor ICT4D Non Formal Education Policy Final Report for Imfundo: Partnership for IT in education.* Department for International Development.

Kaarilahti, N., Pekkola, S., & Pohjola, P. (2006). Towards Formalised End User Participation in Information Systems Development: Bridging the Gap between Participatory Design and ISD Methodologies. *Proceedings of the Ninth Conference on Participatory Design: Expanding boundaries in design. 1*, pp. 21-30. Trento: ACM.

Lachman, R., Nedd, A., & Hinnings, B. (1994). Analyzing Cross National Management and Organizations. *Management Science , 40* (1), 40-55.

Leffingwell, D. (1997). Calculating the Return on Investment from More Effective Requirements Management. *American Programmer , 12* (4), 13-16.

Lewin, K. (1946). Action Research and Minority Problems. *Journal of Social Sciences , 2*, 34-46.

Lui, L., & Kooshabeh, P. (2003). Paper or Interactive? A Study of Prototyping Techniques for Ubiquitous Computing Environments. *Conference on Human Factors in Computing Systems.* Ft Lauderdale.

Maiden, N. A., & Rugg, G. (1996). ACRE: Selecting methods for Requirements Aquisition. *Softwaree Engineering Journal , 11* (3), 183-192.

Maslow, A. (1970). *Motivation and Personality* (2nd ed.). New York: Harper and Row.

Matts, C. (2003). *Negative feedback: Encouraging the "Right Stuff"*. Retrieved January 5, 2009, from Pols Consulting: http://www.pols.co.uk/business-coach/NegativeFeedback.pdf

Mayhew, D. (1999). *The Usasbility Engineering Lifecycle: A Practitioner's Handbook for User Interface Design.* San Fransisco: Morgan Kauffman Publishers.

McNurlin, R. H., & Sprague, B. C. (1993). *Information Systems and Management practice* (3rd ed.). Englewood Cliffs: Prentice Hall International.

Mouton, J. (2001). *How to Succeed in Your Masters and Doctoral Studies: A South African Guide and Reference Book.* Pretoria: Van Schaiks.

Muyingi. (2006). *Research at the Telkom COE for Developmental e-commerce*. Retrieved from Telkom CoE for Developmental e-commerce at Fort hare and Rhodes University: http://cs.ufh.ac.za/coe/research.html

Mwabu, G., & Thorbecke, E. (2001). Rural Development, Economic Growth and Poverty Reduction in Sub Saharan Africa. *Journal of African Economies , 13* (1), i16-i65.

Myers, C. (1997). *The Human Element in Technology Adoption.* Retrieved March 22, 2005, from http://www.stsc.hill.af.mil/crosstalk/1997/12/human.asp

Norton, D. (2003). Rainbow Usability. *ACM Digital Library Interactions , 10* (2), 9-12.

Oates, B. J. *Researching Information Systems and Computing.* London: Sage Publications Ltd.

O'Brien, R. (2001). An overview of the methodological approach of action research. In R. Richardson, *Theory and Practice of Action Research.* João Pessoa: Universidade Federal da Paraíba.

Oppenheim, A. N. (1992). *Questionnaire Design, Interviewing and Attitude Measurement.* London: Pinter.

Pade, C. (2006). *An Investigation of ICT Project Management Techniques for Sustainable ICT Projects in Rural Areas.* Rhodes University, Grahamstown.

Pade, C., Mallinson, C., & Sewry, D. (2006). An Investigation of Sustainable ICT projects in Rural Development. *Proceedings of the 2nd Sangonet Conference on ICTs for CIvil Society.* Johannesburg.

Petrazzini, B., & Kibatti, M. (1999). The Internet in Developing Countries. *Communications of the ACM , 42* (6), 31-36.

Polanyi, M. (1967). *The Tacit Dimension.* New York: Anchor Books.

Rauterberg. (2003). M. In E. Graefe, *Tekom Jahrestagung* (pp. 175-178). Wiesbaden: Gesellschaft fuer technische Kommunikation e.V.

Rettig, M. (1994). Prototyping for Tiny Fingers. *Communications of the ACM , 37* (4), 21-27.

Royce, W. (1970). Managing the Development of Large software Systems. *Proceedings of the IEEE WESCON.*

Rzepka, W. E. (1989). A Requirements Elicitation Engineering Testbed: Concept, Status and First Results. *Proceedings of the Twenty Second Annual Hawaii International Conference on System Sciences* (pp. 339-347). IEEE Computer Society.

Schach, S. R. (1998). *Classical and Object Oriented Software Engineering with UML and C++.* New York: McGraw-Hill.

Schneiderman, B. (1998). *Designing the User Interface* (3rd ed.). Boston: Addison-Wesley.

Scott, K. (2007). *Towards a Strategic Alignment Maturity Assessment Method for Fast Moving Consumer Goods.* Masters Dissertation, Universtity of Fort Hare, East-London.

Seger, C. A. (1994). Implicit Learning. *Psychology Bulletin , 115* (2), 163-196.

Silver, D., & Wood, J. (1995). *Joint Application Development* (2nd ed.). New York: John Wiley.

Smith, M. K. (2003). *Michael Polanyi and Tacit Knowledge*. Retrieved December 15, 2008, from Encyclopaedia of Informal Education: http://www.infed.org/thinkers/polanyi.htm

Svanaes, D., & Seland, G. (2004). Putting the users center stage: role playing and low-fi prototyping enable end users to design mobile systems. *CHI,* (pp. 479-486). Vienna.

Timmermans, H. (2002). Natural Reseource Use at Dwesa-Cwebe. In D. Fay, H. Timmermans, & R. Palmer, *From Conflict to Negotiation.* HSRC Press.

Trust, H. S. (2005). Retrieved August 10, 2005, from http://www.hst.org.za/publications/535

UNDP. (2001). *Making New Technologies Work for Human development.* New York: UN.

Wiegers, K. E. (1996). *Creating a Software Engineering Culture.* New York: Dorset House.

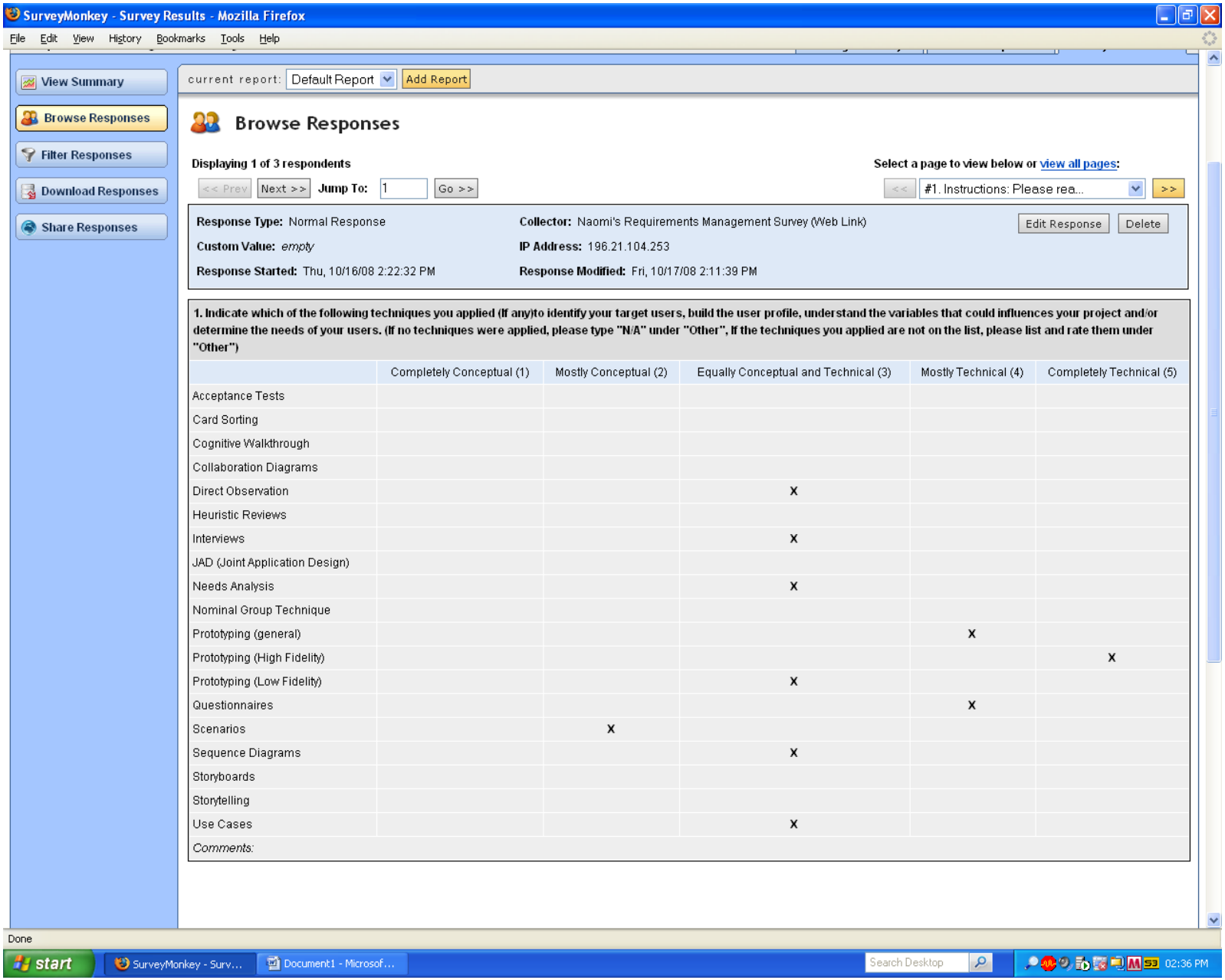
Yin, R. K. (2003). *Case Study Research: Design and Methods* (3rd Edition ed.). California: Sage Publications.

Zake, O. (2007). *An Investigation into the Critical Success Factors of a Rural Telecentre with Emphasis on Knowledge Sharing.* University of Fort Hare, East-London.

Zhang, Z. (2007). Effective Requirements Development- A Comparison of Requirements Elicitation techniques. *Software Quality Management XV; Software Quality in the Knowledge Society* (pp. 225-240). British Computer Society.

Appendices

Appendix A: Sample Survey Response



SurveyMonkey - Survey Results - Mozilla Firefox

File Edit View History Bookmarks Tools Help

View Summary | Browse Responses | Filter Responses | Download Responses | Share Responses

current report: Default Report Add Report

Browse Responses

Displaying 1 of 3 respondents

<< Prev Next >> **Jump To:** 1 Go >>

Select a page to view below or view all pages:

<< #1. Instructions: Please rea... >>

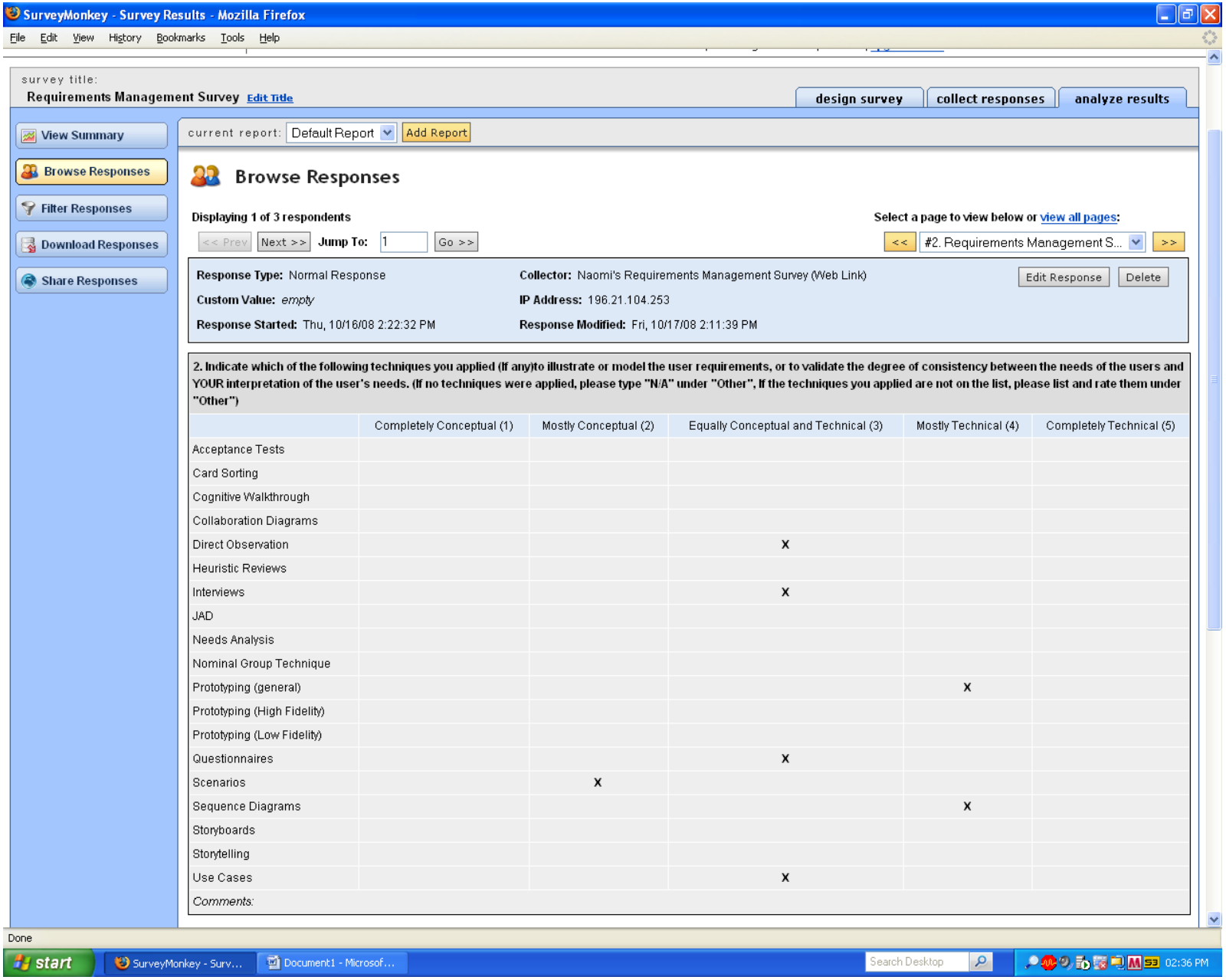
Response Type: Normal Response
Collector: Naomi's Requirements Management Survey (Web Link)
Edit Response Delete
Custom Value: *empty*
IP Address: 196.21.104.253
Response Started: Thu, 10/16/08 2:22:32 PM
Response Modified: Fri, 10/17/08 2:11:39 PM

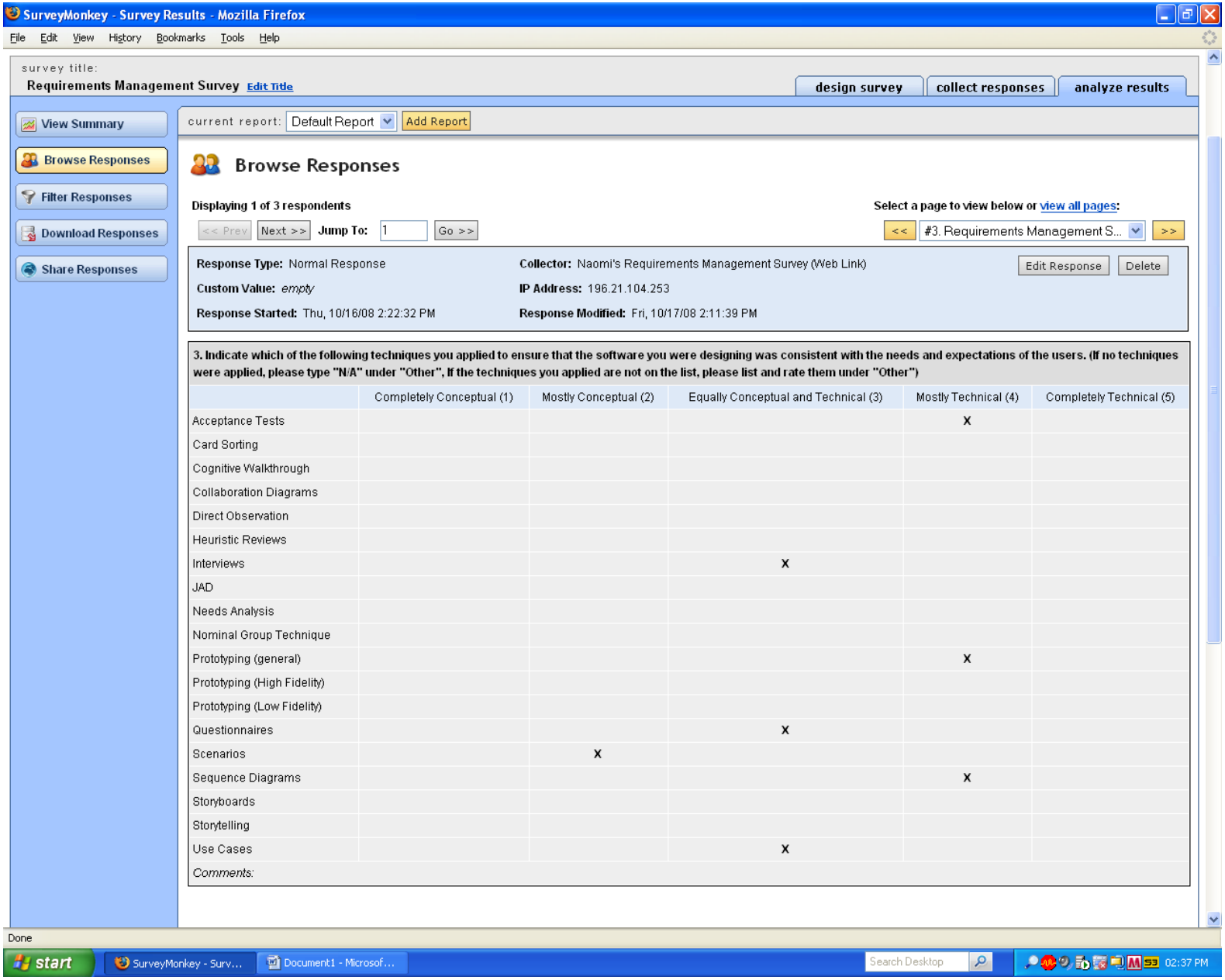
1. Indicate which of the following techniques you applied (If any)to identify your target users, build the user profile, understand the variables that could influences your project and/or determine the needs of your users. (If no techniques were applied, please type "N/A" under "Other", If the techniques you applied are not on the list, please list and rate them under "Other")

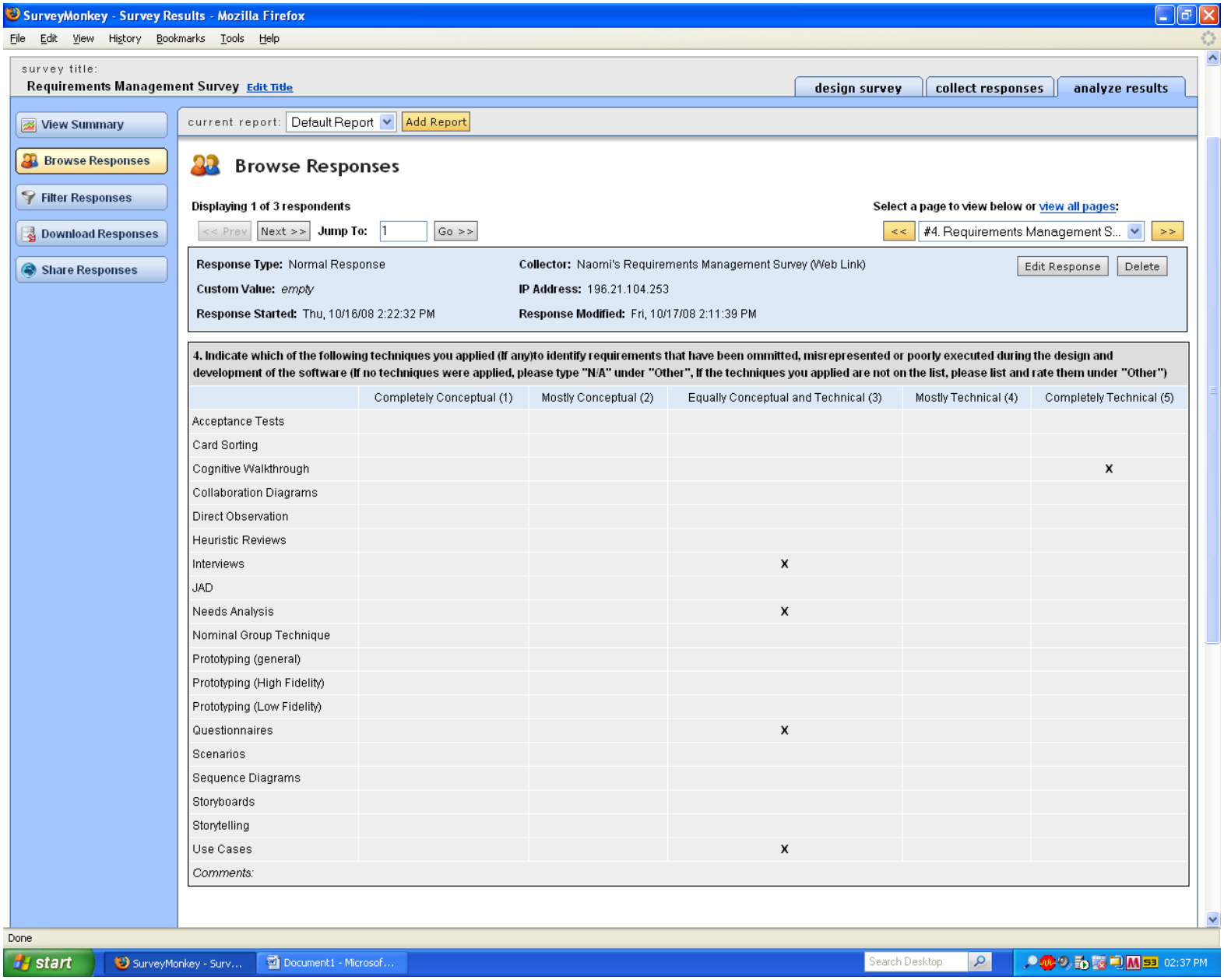
	Completely Conceptual (1)	Mostly Conceptual (2)	Equally Conceptual and Technical (3)	Mostly Technical (4)	Completely Technical (5)
Acceptance Tests					
Card Sorting					
Cognitive Walkthrough					
Collaboration Diagrams					
Direct Observation			X		
Heuristic Reviews					
Interviews			X		
JAD (Joint Application Design)					
Needs Analysis			X		
Nominal Group Technique					
Prototyping (general)				X	
Prototyping (High Fidelity)					X
Prototyping (Low Fidelity)			X		
Questionnaires				X	
Scenarios		X			
Sequence Diagrams			X		
Storyboards					
Storytelling					
Use Cases			X		
Comments:					

Done

start | SurveyMonkey - Surv... | Document1 - Microsof... | Search Desktop | 02:36 PM


SurveyMonkey - Survey Results - Mozilla Firefox
File Edit View History Bookmarks Tools Help
survey title:
Requirements Management Survey Edit Title
design survey
collect responses
analyze results
View Summary
Browse Responses
Filter Responses
Download Responses
Share Responses
current report: Default Report Add Report
Browse Responses
Displaying 1 of 3 respondents
Select a page to view below or view all pages:
<< Prev Next >> Jump To: 1 Go >>
<< #2. Requirements Management S... >>
Response Type: Normal Response
Collector: Naomi's Requirements Management Survey (Web Link)
Edit Response Delete
Custom Value: empty
IP Address: 196.21.104.253
Response Started: Thu, 10/16/08 2:22:32 PM
Response Modified: Fri, 10/17/08 2:11:39 PM
2. Indicate which of the following techniques you applied (If any)to illustrate or model the user requirements, or to validate the degree of consistency between the needs of the users and YOUR interpretation of the user's needs. (If no techniques were applied, please type "N/A" under "Other", If the techniques you applied are not on the list, please list and rate them under "Other")
Completely Conceptual (1)
Mostly Conceptual (2)
Equally Conceptual and Technical (3)
Mostly Technical (4)
Completely Technical (5)
Acceptance Tests
Card Sorting
Cognitive Walkthrough
Collaboration Diagrams
Direct Observation X
Heuristic Reviews
Interviews X
JAD
Needs Analysis
Nominal Group Technique
Prototyping (general) X
Prototyping (High Fidelity)
Prototyping (Low Fidelity)
Questionnaires X
Scenarios X
Sequence Diagrams X
Storyboards
Storytelling
Use Cases X
Comments:
Done
start
SurveyMonkey - Surv...
Document1 - Microsof...
Search Desktop
02:36 PM


SurveyMonkey - Survey Results - Mozilla Firefox
File Edit View History Bookmarks Tools Help
survey title:
Requirements Management Survey Edit Title
design survey
collect responses
analyze results
View Summary
Browse Responses
Filter Responses
Download Responses
Share Responses
current report: Default Report Add Report
Browse Responses
Displaying 1 of 3 respondents
Select a page to view below or view all pages:
<< Prev Next >> Jump To: 1 Go >>
#3. Requirements Management S...
Response Type: Normal Response
Collector: Naomi's Requirements Management Survey (Web Link)
Edit Response Delete
Custom Value: empty
IP Address: 196.21.104.253
Response Started: Thu, 10/16/08 2:22:32 PM
Response Modified: Fri, 10/17/08 2:11:39 PM
3. Indicate which of the following techniques you applied to ensure that the software you were designing was consistent with the needs and expectations of the users. (If no techniques were applied, please type "N/A" under "Other", If the techniques you applied are not on the list, please list and rate them under "Other")
Completely Conceptual (1)
Mostly Conceptual (2)
Equally Conceptual and Technical (3)
Mostly Technical (4)
Completely Technical (5)
Acceptance Tests X
Card Sorting
Cognitive Walkthrough
Collaboration Diagrams
Direct Observation
Heuristic Reviews
Interviews X
JAD
Needs Analysis
Nominal Group Technique
Prototyping (general) X
Prototyping (High Fidelity)
Prototyping (Low Fidelity)
Questionnaires X
Scenarios X
Sequence Diagrams X
Storyboards
Storytelling
Use Cases X
Comments:
Done
start
SurveyMonkey - Surv...
Document1 - Microsof...
Search Desktop
02:37 PM



SurveyMonkey - Survey Results - Mozilla Firefox

File Edit View History Bookmarks Tools Help

survey title:
Requirements Management Survey Edit Title

design survey | collect responses | analyze results

View Summary
Browse Responses
Filter Responses
Download Responses
Share Responses

current report: Default Report Add Report

Browse Responses

Displaying 1 of 3 respondents

<< Prev Next >> Jump To: 1 Go >>

Select a page to view below or view all pages:

<< #4. Requirements Management S... >>

Response Type: Normal Response
Collector: Naomi's Requirements Management Survey (Web Link)
Custom Value: *empty*
IP Address: 196.21.104.253
Response Started: Thu, 10/16/08 2:22:32 PM
Response Modified: Fri, 10/17/08 2:11:39 PM

Edit Response Delete

4. Indicate which of the following techniques you applied (If any)to identify requirements that have been ommitted, misrepresented or poorly executed during the design and development of the software (If no techniques were applied, please type "N/A" under "Other", If the techniques you applied are not on the list, please list and rate them under "Other")

	Completely Conceptual (1)	Mostly Conceptual (2)	Equally Conceptual and Technical (3)	Mostly Technical (4)	Completely Technical (5)
Acceptance Tests					
Card Sorting					
Cognitive Walkthrough					X
Collaboration Diagrams					
Direct Observation					
Heuristic Reviews					
Interviews			X		
JAD					
Needs Analysis			X		
Nominal Group Technique					
Prototyping (general)					
Prototyping (High Fidelity)					
Prototyping (Low Fidelity)					
Questionnaires			X		
Scenarios					
Sequence Diagrams					
Storyboards					
Storytelling					
Use Cases			X		
Comments:					

Done

start | SurveyMonkey - Surv... | Document1 - Microsof... | Search Desktop | 02:37 PM

Appendix B: Summary of Survey Responses

Analysis

	Completely Conceptual	Mostly Conceptual	Equally Conceptual and technical	Mostly Technical	Completely Technical
Interviews	0	2	2	0	0
Prototyping H	0	0	0	0	1
Prototyping L	0	1	1	0	0
Prototyping G	0	0	0	1	0
Direct Observation	0	0	1	0	0
Needs Analysis	0	0	1	0	0
Use Cases	0	0	1	0	0
Questionnaires	0	0	0	1	0
JAD	1	0	0	0	0
Sequence Diagrams	0	0	1	0	0
Scenarios	0	1	0	0	0
Totals	1	4	7	2	1
Weigthed Totals	1.5	4	0	2	1.5

Specification

	Completely Conceptual	Mostly Conceptual	Equally Conceptual and technical	Mostly Technical	Completely Technical
Interviews	0	0	2	0	0
Questionnaires	0	1	1	0	0
JAD	0	0	1	0	0
Prototyping G	0	0	1	2	0
Cognitive Walkthrough	0	0	0	0	1
Direct Observation	0	0	2	0	0
Sequence Diagrams	0	0	0	1	0
Use Cases	0	0	1	0	0
Scenarios	0	1	0	0	0
Totals	0	2	8	3	1
Weighted Totals	0	2	0	3	1.5

Design

	Completely Conceptual	Mostly Conceptual	Equally Conceptual and Technical	Mostly Technical	Completely Technical
Interviews	0	1	1	0	0
Questionnaires	0	0	1	0	0
Prototyping G	0	0	1	1	0
Direct Observation	0	0	2	0	0
Sequence Diagrams	0	0	0	2	0

Use Cases	0	0	1	0	0
Scenarios	0	2	0	0	0
Acceptance Tests	0	0	0	1	0
Totals	0	3	6	4	0
Weighted Totals	0	3	0	4	0

Implementation

	Completely Conceptual	**Mostly Conceptual**	**Equally Conceptual and Technical**	**Mostly Technical**	**Completely Technical**
Interviews	0	1	2	0	0
Questionnaires	0	0	2	0	0
Needs Analysis	0	0	2	0	0
Cognitive Walkthrough	0	0	0	0	2
Direct Observation	0	0	3	0	0
Use Cases	0	0	3	0	0
Total	0	1	12	0	2
Weigthed Total	0	1.5	0	0	4

Operation

	Completely Conceptual	**Mostly Conceptual**	**Equally Conceptual and Technical**	**Mostly Technical**	**Completely Technical**
Interviews	0	1	1	0	0
Questionnaires	1	0	2	0	0
Prototyping (general)	0	0	0	1	1
Acceptance Tests	0	0	0	1	0
Direct Observation	0	0	2	1	0
Use Cases	0	0	0	1	0
Totals	1	1	5	4	1
Weighted Totals	1.5	1	0	4	1.5

Appendix C: First Extract of Dialogue Representing Analytic Techniques

Researcher 1 to All Oct 16

Hi there,

We just returned from our final data collection stage of the baseline study trip. It was really interesting to hear what the community had to say, and we have some interesting results which the group is preparing to put together as soon as we can. We are left with the migration interviews, and then we are on our way.

I made a few observations and learned a few lessons, which we will share with you in due course once we have finalised our report.

I would like to plead with all those involved in starting new projects or adding to the platform etc. You need to understand that the people in Dwesa are human beings, have feelings, get excited, and also get disappointed. As one of the project members did indicate, there have been too many people coming in with questionnaires to fulfil their personal research interests, but have left the community with expectations and promises that are not met. So, in order to not jeopardise the project, please be considerate and careful with your research approaches and promising words, which can be misleading sometimes. I'm not just making this up, but this was brought up in two of the focus groups. For instance, the craft ladies have not seen, or do not even know that there is a Dwesa website. They felt that they were promised things and given amazing ideas but ALAS, they have not seen the results. One then wonders why they have stopped coming for training. As was mentioned earlier, the school is not the only target. Also, the baseline study can give you a view of the current status and needs in the community, you are interested. Please respect their way of doing things, their culture, and their expectations.

Then also, another concern that some youth brought up is that there is no training structure. Yes, one probably assumed that they would be able to come up with one themselves, but hmmmm that is asking them for too much now. They need guidance to come up with a training structure in the community, and also importantly, an assessment procedure. I have offered to help her come up with one, but I would also appreciate help, from other project team members, who may have done this before. Proof of ability is very important, hence the need for certificates to show they have some computer literacy. Imagine if we didn't have a certificate with our degrees, what's the point if people wont believe your ability when you look for a job.

Kind Regards,

Researcher 1

Researcher 2 Oct 16 to A

Hi, very important observations I think Researcher 1, maybe one reason for this is the many "research focused" projects carried out this year, baseline, Village Connection study etc., but some of this must be related to the actual RandD projects too..

I think it is important to have sound principles, procedures and methodologies in place for community engagement and interaction aspects in these projects. People concerned, the users, must be part of the process, not only as objects of research or input in the beginning, but also as co-developers of the solutions? Maybe in the future these methodologies should be more formalised and standardised in some way across the many projects that people are engaged with. Could also the work be done more "in situ", where the people actually are, show them the progress, prototypes, problems, and gather more regular and active input?

I also think the new ways of engaging the community in the research / development processes must be transparent, so that the community members realise that it's not always that straightforward, and sometimes development comes to a halt for many different reasons. In some cases, they must expect that the readymade solution just lands on their feet some day, rather sooner than later, and when that does not happen, get anxious about the progress?

Rgds
Researcher 2

Researcher 3 Oct 16

to A, A

Hi,

my comments embedded.

Hi there,

We just returned from our final data collection stage of the baseline study trip. It was really interesting to hear what the community had to say, and we have some interesting results which the group is preparing to put together as soon as we can. We are left with the migration interviews, and then we are on our way.

I made a few observations and learned a few lessons, which we will share with you in due course once we have finalized our report.

I would like to plead with all those involved in starting new projects or adding to the platform etc. You need to understand that the people in Dwesa are human beings, have feelings, get excited, and also get disappointed. As Robin did indicate, there have been too many people coming in with questionnaires to fulfil their personal research interests, but have left the community with expectations and promises

that are not met. So, in order to not jeopardise the project, please be considerate and careful with your research approaches and promising words, which can be misleading sometimes. I'm not just making this up, but this was brought up in two of the focus groups. For instance, the craft ladies have not seen, or do not even know that there is a Dwesa website.

This is really even shocking for me! For example One of the software developers has been interacting with those women, or not? I am surprised to hear that. Even Naomi reports in her paper (...) how less input was gathered from the community on a set of demonstrations of website prototype run by a team...

They felt that they were promised things and given amazing ideas but ALAS, they have not seen the results. One then wonders why they have stopped coming for training.

I think, as highlighted by Researcher 2, we have to put clear procedures on how working with the community and reporting back the progress. "they have stopped coming for training..." never been warned at my best knowledge.

As was mentioned earlier, the school is not the only target. Also, the baseline study can give you a view of the current status and needs in the community, you are interested. Please respect their way of doing things, their culture, and their expectations.

We have to work hard here given the fact that our students are more "science" or "tech" oriented than those in IS or humanity discipline and as such need continuous briefing on the " human issues" site of the project. For example, the way translators (Xhosa/English) interact, and any informal interaction, conversations, private talk with a folk may be unconsciously building bias. One way to help here maybe keeping continuous face to face (costly!)briefing meeting (but mailing list does it already well!) on story, experienced faced by team members every time they come back from the site.

Then also, another concern that some youth brought up is that there is no training structure. Yes, one probably assumed that they would be able to come up with one themselves, but hmmmm that is asking them for too much now. They need guidance to come up with a training structure in the community, and also importantly, an assessment procedure. I have offered to help her come up with one, but I would also appreciate help, from other project team members, who may have done this before. Proof of ability is very important, hence the need for certificates to show they have some computer literacy. Imagine if we didn't have a certificate with our degrees, what's the point if people wont believe your ability when you look for a job.

Kind Regards,

Researcher 1

Researcher 4 to A

Hi all,

I agree completely with your concerns, and have been addressing it in my write up. One of the biggest problems is that there is very little integration between the research projects. We are operating quite independently, and yet we probably should be informing each others research. It was mentioned to me (quite some time back), that although I was looking at developing a requirements elicitation methodology for the E-Mall, my findings were meant to influence consequent versions of the software. This made sense especially seeing that the software developers and I progressed at different paces. It is unfortunate that the crafts ladies are disappointed. I'm sure we can use our experiences and knowledge to manage the process in a better way from here forward.

I have developed a methodology that specifically addresses eliciting and managing requirements throughout the software development lifecycle, and would really like to get involved in devising a strategy to overcome some of the shortcomings of our previous efforts.

Regards,

Researcher 4

Researcher 5 Oct 16 to A

Hi Researcher 1,

I look forward to hearing your conclusions and experiences. I would be interested to know if our Village Connection research was also mentioned while you were there? It was very important for us to avoid creating expectations, and we tried to make every effort to emphasize that there was a very real possibility that nothing would come of our research. But we are also learning and for our own process development, it would be good to know whether that message came across or not.

At the same time, I think it's nearly impossible to research a concept without creating some expectations that it will come about (unless your research is very oblique). Asking people to visualise what it would be like to own or be a part of something already creates a sense of ownership and desire, no matter what you say about its likelihood of being implemented. As Researcher 2 says, the best way to approach this is perhaps not to avoid creating expectations, but to give the community a stake in the outcome through a co-design and co-development approach. That way they have a hand in bringing it about but also are in a better position to understand if it can't be realised.

I found there was a general sense in the community that opportunities come from outside and on the terms set by outsiders (note that this was only my impression and I can't claim to understand the people of Dwesa based on our very brief experience there). I would hope that they can gain a greater sense of power over what happens in their community, and play a greater role in shaping how opportunities are presented. This can happen if they are involved as equal partners in the process from start to finish. The Living Labs concept as I understand it is a great platform for building this relationship and I hope that it will be successful. I was already highly impressed by the relationship that exists there already.

I am personally very interested in the processes of co-design and development with rural communities and have come across a few projects internationally that are developing (or have developed) methodologies for this type of engagement. If anyone knows of others, I'd be keen to hear about them.

The Center for Sustainable Global Enterprise at Cornell University has developed the Base of the Pyramid Protocol (2nd ed), which presents a methodology for community partnership around innovation. I don't know that their methodology is appropriate in its entirety, but there are elements that might be useful. They are apparently producing a handbook for applying the protocol which will hopefully make the methodologies clearer. The protocol (not the handbook) can be downloaded from http://tinyurl.com/53876o

Philips have also followed a co-design and co-creation approach in producing a smokeless stove for Indian villagers (http://tinyurl.com/46g69c).

But I think that the long-term commitment and close relationship with Dwesa presents opportunities to develop new methodologies that are uniquely appropriate there and that might inform the processes used elsewhere as well. I also think many of you already have a wealth of experience in engaging with rural communities, and I value the insights that come through this list. It is also valuable to hear directly from the champions what works and what doesn't.

Kind regards,
Researcher 5

Researcher 6 Oct 16 to A

Hi Researcher 1,

am looking forward to the report!

To your mail:
1. A promise we make must be a promise we keep. Since research is about trying out things that promise a lot but are often not built to the end, I think this might be difficult for students. And yes, feedback is important. Who is responsible for that? There should be a champion that the community can approach to ask such questions to clear them up.

2. Training: Lorenzo is working on a training structure and an accreditation (testing) so that the project can start giving certified ICT training by middle of next year (lorenzo?).

3. Since one of our key champions at Mpume left, I see our youth champion kind of as a 'lone champion'. The others seem to use the system to their own ends. We need more champions to be identified and cultivated. Or am I seeing this wrongly?

Best regards, Researcher 6

Researcher 7 Oct 16 to A

I read Researcher 1's despatch with great interest and pleasure. The people in Dwesa and surroundings are indeed fortunate to have her. The stereoptype of IT and IS professionals and students is that they care more about hardware, software and systems than people, but it simply isn't true. Nor is a caring approach limited to Researcher 1: members of this list have evidence of concern for the sensitivities of the residents from so many of our 'conversations'. The key is of course the frequent field trips with the opportunities these offer to enter into relationships of trust with local people who then also voice the concerns of others both within deliberate set-ups such as the focus groups but also informally. Provided that the project then responds to these concerns with constructive solutions, the prospects are good that the Siyakula LL will not be just another cynical and 'extractive' research project under another name, but that it will encapsulate the spirit of the 'Living Lab' concept with a truly participatory and constructive approach.

Researcher 7

Researcher 7 Oct 16 to A

Re Researcher 6's points:

> 1. A promise we make must be a promise we keep. Since research is about
> trying out things that promise a lot but are often not built to the end,
> I think this might be difficult for students. And yes, feedback is

> important. Who is responsible for that? There should be a champion that
> the community can approach to ask such questions to clear them up.

Early in 2005 I went up to Dwesa-Cwebe to report back to 'the community' on a project the ISER had been running there in 2003-4. I was on the Steering Committee, not an active researcher, but I thought it important to take the place of the Project Leader after he left prematurely and share the results locally. We had two meetings, one on each side of the Mbashe, that a local man organised for us, for pay. He didn't do a great job and the attendance wasn't large (about 20) in either case. But one thing that was said at both meetings, accompanied by much nodding of heads, was that it was rare to have any report back at all from researchers, and they really appreciated the opportunity, thanking us kindly for taking the trouble. Obviously, the LL being long-term, we shouldn't wait till the end, but report back at intervals. Providing especially hot food and cool drinks is greatly appreciated!

> 2. Training: One of our researchers is working on a training structure and an
> accreditation (testing) so that the project can start giving certified
> ICT training by middle of next year.

That should help the youth in their search for employment.

Since one of our key champions at Mpume left, I see our youth champion kind of as a 'lone champion'. The others seem to use the system to their own ends. We need more champions to be identified and cultivated. Or am I seeing this wrongly?

At the very least one needs one per school/village, but so far one only hears of our youth champion, it is true.

Researcher 1 Oct 17 to A

Hi,

Thank you for your advise about how to best work with the community. I do hope that people can take things seriously and consider these views. Once upon a time we used to have meetings to discuss the project progress (with Rhodes and Fort Hare). The only project team meetings we have had are the Baseline study ones with Fort Hare, to my knowledge. Unless maybe it's just the Rhodes side. I understand and got the impression people are not so keen with formalities, but in all honesty, it keeps us on track. I hardly know the new people involved or who have taken up a Masters on the Fort Hare side, and what they are researching on. It just happens that occasionally, now only towards the end of the year, I hear there is someone developing a Judiciary system, for instance. Bottom line is, as Naomi pointed out, there is no collaboration between the research projects, which really is a challenge, especially when we are left with having to convince the people in the community that the project can have an impact, because the Baseline study team is the one that has interacted with them to find out their views. A good thing to do, is to update the wiki for instance.

Just to clarify, we do have another teacher as an active project champion at Mpume. I think she is also the chair of the committee they formed. She also mentioned the need for a training structure. Ok, I guess the accreditation aspect can come a bit later, but a training structure is also essential. Perhaps, before middle of next year, one could discuss with them a procedure for training, because at the moment it is pretty random and sporadic. Maybe for the moment an informal structure, which could build up onto the formal one, just to provide the trainers with some guidance.

Regards,

Researcher 1

Appendix D: Second Extract from Dialogue Representing Analytic Techniques

Researcher 1 show details 6/13/07

to A

Hi guys,

I was just checking out our e-mall and was pretty impressed with it. I definitely like the new version too (it looks pretty slick) :)

I just had one suggestion about the descriptions of the goods. Perhaps we can give a more comprehensive description than the one-liners we have now?

For example in the arts and craft section, the first item is an "intsaka-Necklace" for R100 with a description of "Necklace, you wear it in any occasion".

Perhaps we can change that to
Item name: Intsaka (necklace)
Description: African beads in sand and amber tones. This intricately patterned necklace is 5cm wide and ... Traditionally, it is worn ...
Its cultural significance...

I personally don't know any "cultural" info about the jewellery, but you get the idea hey? Maybe something similar to the information in the national geographic shop..
(http://shop.nationalgeographic.com/gateway/1020/116.html)

Regards,

Researcher 1

Researcher 2 show details 6/13/07

to A

Hi Researcher 1
That is an excellent idea, i think that would make it even more interesting for the buyers. It also provides more info on how and where it can be worn and how it was made, which is good even from the cultural point of view.

thanks
Researcher 2

Researcher 3 show details 6/13/07

to A

HI Researcher 1 and Researcher 2, good idea indeed.

How should the thing be organized practically? We need somebody that knows the products and their context, somebody with a flair for writing and somebody with a feel for selling. Any volunteer? (We would like to open up shop in a few days.)

Researcher 3

Researcher 4 show details 6/13/07

to coedwesa-l

Hi all

Mpho Molapisi is in charge of the anthropology collections at the Albany Museum and is very interested in black African material culture of all kinds (m.molapisi@ru.ac.za). He might be able to help in describing the items, though it would also be good to interview the makers of the crafts about the background and context of their creations. Boosting the items is of course a different kind of skill, but it needs to be based on what local people or Mpho can provide.

Researcher 4

Appendix E: Publications Resulting from This Research

A Model for Eliciting User Requirements Specific to South African Rural Areas

Naomi Isabirye
University of Fort Hare
P. O. Box 7426
East London, 5200
+27 (0)43 722 3023
nisabirye@gmail.com

Stephen Flowerday
University of Fort Hare
P. O. Box 7426
East London, 5200
+27 (0)43 7047071
sflowerday@ufh.ac.za

ABSTRACT

Dwesa is a rural town situated in South Africa's Eastern Cape Province. Its residents are plagued with the challenges of poverty, poor access to public services, unemployment and low levels of literacy. A project is currently underway in Dwesa to develop an E-Commerce platform that will 'connect' the residents with the necessary access to services and markets by providing them with the necessary tools in order to help in alleviating some of their challenges. The purpose of this paper is to evaluate existing requirements elicitation techniques and methodologies in the context of rural information technology implementations. The primary objective is to develop a model for user requirements elicitation in Dwesa. Many requirements elicitation techniques and frameworks exist, but few have been evaluated in the context of rural software implementations. Furthermore, requirements elicitation techniques should not be applied simply as steps to gather information. Instead this paper proposes a model that can be applied to assimilate the contribution of knowledge regarding the stakeholders, problem and solution characteristics, and other characteristics into the software development process for the effective elicitation of requirements.

Categories and Subject Descriptors

D.2.1 Requirements/Specifications: Requirements Elicitation Techniques and Methodologies

General Terms

Management, Design

Keywords

Requirements Elicitation, Elicitation Methodologies, Elicitation Techniques, Rural Software Application

1. INTRODUCTION

Dwesa is a rural village in the Eastern-Cape province of South Africa. Dwesa's rich cultural and political heritage has resulted in it being the subject of extensive social research. Research has revealed that Dwesa's ecological diversity has caused it to be a highly protected area in terms of conservation policies [6]. Unfortunately, the consequences of historical injustices and innumerable conservation laws surrounding the use of natural resources in Dwesa have grossly incapacitated the residents [6]. Dwesa's residents are critically impoverished, and are restricted by poor access roads, lack of telecommunications infrastructure, electricity and basic public services. Entrepreneurs in Dwesa are therefore limited to marketing their goods or services within a small geographical region. An e-commerce platform has the potential to expand the boundaries of their market, and therefore, boost the economic activity within their society.

As part of a project to deploy an E-Commerce platform in the area, a study was conducted to evaluate the effectiveness of various requirements elicitation techniques in a rural context. The study was founded on the review of literature surrounding requirements elicitation techniques and experience applying these techniques in Dwesa. The study evaluated individual and group interviews as well as prototyping as requirements elicitation techniques. Individual and group interviews allowed the analyst to gain an insight into the business activities taking place in the area, and to identify project champions from the community to support the project. The main challenges observed were the language barriers between the interviewers and community members, as well as the unbalanced contribution of opinions during the group interviews.

The prototype demonstration allowed the developer to upload images of the crafts to be sold through the E-Commerce platform, and to obtain descriptions and indigenous names of the items. However, the major drawback of prototyping was that the audience did not offer any criticism regarding the technical or aesthetic aspects of the prototype.

The study concluded that although there is value in critically evaluating requirements elicitation techniques, the quality of the requirements elicitation process could benefit even more from the establishment of a model for requirements elicitation.

An evaluation of two requirements elicitation methodologies is conducted in this paper. The first model, Browne and Rogich's Task Characteristics Model [3], is founded on the application of procedural and substantive prompts to elicit information from users of a system. The second methodology, Davies and Hickey's Unified Model of Requirements Elicitation [4], is based on the principle that prior to elicitation of requirements, a technique selection process must take place based on the knowledge of the

SAICSIT 2008, 6 - 8 October 2008, Wilderness Beach Hotel, Wilderness, South Africa

problem and solution domain, the project characteristics, and the situational characteristics.

The purpose of the study was to contextualize the techniques and methodologies for the development of the E-Commerce platform being deployed in Dwesa. This paper aims at summarizing the findings of the study in order to propose a requirements elicitation model applicable to the development of software in rural areas.

2. REQUIREMENTS PHASE OF SOFTWARE DEVELOPMENT

The requirements phase is broadly recognized as one of the most critical and challenging phases of systems development and has been identified as one of the major causes of failures in software development projects [2, 12]. Between 40% and 60% of errors in software projects can be attributed to errors in the requirements phase [8, 12]. Furthermore, ambiguous user requirements can result in revisions amounting to as much as 40% of the total software development costs, while between 70 and 85% of revisions to software development projects result from requirements errors [8, 12]. A study [1, 12] found that it costs 68 times more to make corrections to a software product after it has been in operation, than if it were to be corrected during the requirements phase. It has also been found that the five most frequently reported causes of software cost estimation relate to requirements engineering [5, 12]. Although these figures are based on isolated studies, and do not necessarily reflect a universal truth, they are all conducted by highly recognized authors in the field of requirements management, and they do highlight the importance of the requirements phase. Unfortunately, software development teams are almost always restricted in terms of time, and hence, often focus more on the technical aspects of software development at the expense of the requirements phase. Emphasis should be placed on early stage quality practices because a good quality requirements process has a high leveraging effect on the outcome of the software development process [12].

The ultimate goal of software development is to create software systems that, when implemented, solve users' problems, leverage their opportunities, or satisfy their needs [4, 12]. While software development may follow a broad range of different methodologies or frameworks, requirements activities tend to be performed at the beginning of the software development lifecycle, and yet as the needs of the users or the organization change, so too must the requirements [4]. This must be accommodated in the software development methodologies.

3. ELICITING USER REQUIREMENTS IN A RURAL SOUTH AFRICAN SETTING

A study was undertaken to evaluate the process of requirements elicitation based on a literature review and the application of three specific elicitation techniques: individual interviews, group interviews, and prototyping. Individual and group interviews were held with community members with the aim of identifying the entrepreneurial activities taking place in the area, and establishing whether the entrepreneurs were willing to participate in the project. Later, a prototype of the proposed system was demonstrated. The prototype exhibited the basic functionality of any standard E-Commerce website and served the purpose of demonstrating the concept of E-Commerce to the users as well as identifying aspects of the system that the users did not understand.

One of the biggest obstacles faced during the interviews was the language barrier between the interviewer and the community members. The situation in Dwesa is such that few of the entrepreneurs speak English. Amongst the group that is setting up the E-Commerce platform, hardly any of them speak Xhosa, (the language spoken in the Dwesa area). This complicated the interviewing process as the information gained through interviews passed through a translation process where some of the terms or concepts were very difficult to translate. With the assistance of a translator, the interviews were quite effective in exposing the business activities taking place within the community, the individuals involved in the various business activities, the challenges faced by entrepreneurs, and the level of willingness to participate in an E-commerce project to market the goods and services of the various businesses.

Group elicitation techniques generally produce more and improved solutions to problems, utilize resources efficiently, eliminate inferior ideas more effectively, increase productivity, result in superior judgments, have a faster learning curve and ensure tenacity of purpose [11, 7]. Unfortunately groups are subject to many dysfunctions many of which can be resolved through effective facilitation [11]. A broad variety of skills are required, the most important being based on the ability to effectively communicate.

The group interviews were aimed at establishing the degree to which the entrepreneurs were willing to become involved in the project, and to explore ways in which the E-Commerce platform could assist their businesses. An imbalance of contribution by the participants is a common disadvantage of group elicitation techniques [11]. This imbalance was also observed during the group interviews in Dwesa. However, the existence of dominant participants aided the project team in identifying potential project champions from the community. During one of the group interviews, the participants took the initiative to nominate and appoint a community member to serve as their representative amongst the project stakeholders.

A prototype of the proposed E-Commerce software was demonstrated to the entrepreneurs in Dwesa. The prototype allowed the developer to observe omissions or flaws in the system. The developer was successful in verifying the Xhosa translations of items being sold on the system. He was also able to upload images of the crafts that would be sold on the website, and allocate prices to the items. This was useful as it gave the prospective users an indication of what kind of functions the system would offer. Unfortunately, the developer's observations were limited as no criticism was offered by the users, who had no prior exposure to similar technologies, and hence had nothing with which to compare the prototype. The developer relied solely on his own perception as to how the entrepreneurs responded to the system. Prototyping in this scenario provided less value as an elicitation process as there was little feedback obtained from the audience.

During the observation of the prototyping process, it was noted that much time and effort had gone into the development of the prototype. This created the possibility of resistance from the software developer to any proposed changes. Programmers sometimes become attached to their work and as a result, they sometimes fail to acknowledge their mistakes [10]. This is one of the difficulties in prototyping, specifically when a lot of work has been put into the development of the prototype. The prototype

demonstrated in Dwesa took approximately ten months to develop as it was part of a research project. If the prospective users had criticized the prototype, the developer may probably have defended it rather subjectively. This is a scenario identified by certain schools of thought that distinguish between two different approaches to prototyping: high fidelity (Hi-Fi) prototyping and low fidelity (Lo-Fi) prototyping [9]. Instances where much time, effort and resources are engaged in developing a prototype, such as the prototype designed for the Dwesa entrepreneurs, are referred to as Hi-Fi prototyping. This often involves the use of high level prototyping tools and the delivery of semi-functional applications that take weeks to develop. Lo-Fi Prototyping on the other hand, involves the designing of the prototype on paper, testing it with a selection of users and simulating the way it would be used in reality. In the context of the prospective users in Dwesa, applying Lo-Fi prototyping in preference to Hi-Fi prototyping, based on the short amount of time involved, would be at the expense of the high level of interaction and visual presentation that Hi-Fi prototypes provide. It was thus concluded that the two methods can be applied complimentarily, with Lo-Fi prototyping during the earlier stages of development, to aid the developer in defining the requirements, and Hi-Fi prototyping at a later stage to confirm that the user's expectations match the analyst's interpretations.

Based on the evaluation of requirements elicitation technique literature and experience applying various techniques in a rural setting, it was concluded that assessing user requirements elicitation techniques could provide analysts with an understanding of how the techniques can be applied, and could guide analysts to understanding the contexts within which specific techniques could be applicable. However, assessing requirements elicitation methodologies is of greater value than evaluating the requirements elicitation techniques themselves, as models provide guidelines or principles that can be adapted to both the characteristics of the situation for which they are intended to be applied, as well as to the people using the approach.

4. EXPLORING REQUIREMENTS ELICITATION MODELS

Of particular relevance to this paper are Browne and Rogich's Task Characteristics Model plus Davies and Hickey's Unified Model of Requirements Elicitation. These models are founded on the recognition of the relationship, at a broad level, between the problem domain, solution domain, and the users and analysts.

4.1 Browne and Rogich's Task Characteristics Model

Literature surrounding requirements elicitation has been largely centered on guidelines by experts or academics for applying or improving specific requirements elicitation techniques. However, there are very few models that illustrate the process of requirements elicitation in general [3]. The Task Characteristics Model [3] provides a framework that generally models the elements of requirements elicitation and provides the analyst with a framework for determining what the requirements are and therefore, enables them to understand what their goals should be. The strength of the model is the manner in which it perceives the requirements as more than mere documentation of data elements in a business process and recognizes the role requirements elicitation plays in the identification of goals, assumptions, opinions, and desires of users [3].

4.1.1 Elements of the Task Characteristics Model

The main focus of this model is to define a technique to support the interaction of the user and analyst's problem space. The model prescribes the use of both procedural and substantive prompts.

4.1.1.1 Stimulus

According to the Task Characteristics Model, the need for systems development is typically stimulated by the existence of an organizational problem or opportunity. The model categorizes the problems or opportunities into four classes: organizational goals, business processes, tasks that must be performed within the processes to achieve the goals, and information necessary to inform tasks. The model represents these classes in a pyramid structure for two reasons. Firstly, the hierarchical representation places general goals at the top and specific information at the bottom. Secondly, it is based on the assumption that in any systems development initiative, there will be fewer goals than processes, fewer processes than tasks, and fewer tasks than information. Thus, the pyramid structure is a useful means of organizing and describing types of requirements. The hierarchy of goals, processes, tasks and information represents the problem space, which is conceptualized from two problem spaces: the user's problem space and the analyst's problem space.

4.1.1.2 User's Problem Space

During the elicitation of requirements, the user's problem space contains information recalled from long-term memory in response to needs that exist in working memory. The process of eliciting requirements is inhibited by factors such as failure to recall information from long-term memory; cognitive biases and inabilities to articulate routinized procedures that have become automated. Furthermore, the users are faced with both the challenge of recalling details of the current environment, as well as the challenge of envisioning the future preferred organizational environment [3].

4.1.1.3 Analyst's Problem Space

According to the Task Characteristics Model, the successful gathering of requirements is dependent on both the conceptualization of the problem, as well as the visualization of a preferred future organizational environment from the user. This process results in the organizing of the user's responses to create a conceptualization of current and future states of the organization by the analyst. As the analyst begins to build an understanding of the user's needs, partial solutions to the problem are defined. These partial solutions become subject to extensive mental simulations and reformulations that are eventually terminated by the analyst's use of evaluative stopping rules. At this point, the analyst feels he has obtained enough information and thus captures the information in a requirements document [3].

4.1.1.4 Interaction Between User's Problem Space and Analyst's Problem Space

Analysts make use of various types of prompts to stimulate the user's thought processes in order to improve their understanding of the problem and to identify missing requirements. The Task Characteristics Model is founded on the evaluation of two context independent prompting techniques, procedural prompts and substantive prompts. Both types of prompts serve the intention of aiding users and analysts in overcoming cognitive challenges associated with the requirements elicitation task.

i. Procedural prompts

Procedural prompts are designed to elicit information that might otherwise not be obtained due to cognitive challenges and are aimed at prompting users to re-examine their understanding of the system and to scrutinize it critically. Such prompts have been shown to reduce judgmental bias [3]. Furthermore, asking users to summarize their opinions can help overcome the limitations of short-term memory. Procedural prompts may also ask a user to create a scenario describing how he performs a task. Such prompts have been shown to help people articulate information that they would struggle to evoke [3].

ii. Substantive prompts

Substantive prompts are aimed at eliciting specific types of requirements. Substantive prompts should stimulate users' memory structures in a manner that will improve their ability to recall knowledge relevant to system requirements.

4.1.1.5 The Requirements Document

The requirements document includes information elicited from users and gathered from other sources and represents a description of an improved state of a system that is aimed at enabling the organization to achieve its goals [3].

4.1.2 Strengths of Browne and Rogich's Task Characteristics Model

The Task Characteristics Model was empirically evaluated in a comparison between two other well established questioning techniques (interrogatories and semantic methods) [3]. The study found that the Task Characteristics prompts elicited significantly more requirements than both the semantic prompts and the syntactic prompts.

The Following four points are the strengths of the Task Characteristics Model [3]:

i. It offers analysts a clear description of the elicitation task, and aids practitioners in understanding both the problems and opportunities typically found in the relationship between user and analyst.
ii. It provides analysts with a prompting technique for eliciting requirements that is based on theoretical knowledge from several domains.
iii. It is empirically tested and has been found to be effective in eliciting more requirements than other prompting techniques.
iv. It has resulted in an enhanced taxonomy of generic requirement categories for organizing information gathered in the analysis stage of system development that can guide practitioners in the identification and capturing of requirements.

4.1.3 Weaknesses of Browne and Rogich's Task Characteristics Model

The model is based on the findings of a study that was conducted within a single organization with well established business processes. Moreover, one of the criteria for selection of participants for the study was that they must have used a computer system as part of their job function and have had experience using databases. The study tested the prompting techniques as opposed to the model. Thus, the prompting techniques were only evaluated against other prompting techniques as opposed to being tested against a range of requirements elicitation methods. The study proved that the model is effective in eliciting more requirements, but failed to prove that the quality of the requirements elicited were superior. The quantity of requirements obtained does not necessarily imply the success of the requirements elicitation task.

These criticisms of the Task Characteristics Model are only intended to point out the fact that the model has been proven to be effective only under very specific conditions. Unfortunately, when considering the model in a rural scenario, where the business processes have not yet been formerly established, the system is a completely foreign concept and the users have no prior experience with similar systems. Therefore, such a model would be very difficult to implement. The prescription of specific prompting techniques adds an undesirable rigidity to the model in situations where limited knowledge of the various problem spaces exists.

4.2 Davies and Hickey's Unified Model of Requirements Elicitation

The Unified Model of Requirements Elicitation [4] is based on the principle that the requirements elicitation process should involve iterative steps to select an appropriate technique given the profile of the stakeholders, problem domain, and project details. The approach is defined as "knowledge intensive" [4] as it is founded on the principles that adequate knowledge of the environment in which the technology is being implemented is required in order to effectively elicit the user requirements. This knowledge refers to knowledge of the users and stakeholders, the problem and solution, and the project domain.

4.2.1 Strengths of the Unified Model of Requirements Elicitation

The strength of the model is that it explicitly highlights the role knowledge plays in performing both elicitation and elicitation technique selection [4]. The model also provides a unified framework for understanding the purpose and role of requirements elicitation in software development and shows what assumptions existing elicitation methodologies make about the situation. Furthermore, the model makes it easier to tailor existing methodologies for unique situations and allows analysts to create new elicitation methodologies by defining situational characteristics and then observing plus recording the resultant instances of methodologies.

4.2.2 Weaknesses of the Unified Model of Requirements Elicitation

The model is based on the fact that the choice of requirements elicitation techniques is highly dependent on the knowledge of the environment within which the software is applied. It explicitly illustrates that this knowledge includes knowledge of the problem and solution domain, the project characteristic, and states that knowledge of the users, stakeholders and customers contributes to the elicitation process. However, the model does not make any reference to how knowledge of the users can affect the elicitation technique selection process. This is the first shortcoming of the model.

It is possible that in certain instances the profile of the users and stakeholders will have no bearing on the technique selection process. Moreover, taking rural Information and Communication Technology (ICT) users into consideration, it is necessary to include the user and stakeholder profiles as one of the factors that determine the choice of elicitation techniques. Earlier the study explored various requirements elicitation techniques and discussed their application in a rural context. The main characteristic that differentiates rural ICT implementations from urban ICT implementations is the users. Rural ICT use is often challenged by issues such as low literacy levels, little or no prior exposure to technology and the inability to effectively communicate in any language other than their mother tongue. These factors almost certainly affect the choice of elicitation techniques. Another shortcoming of the model is that it does not take into consideration the software development stage in which the techniques will be applied. This is also likely to have a bearing on the effectiveness of certain techniques. Therefore the main limitation of the model is that it attempts to identify the factors that affect the technique selection process and yet it is not possible to exhaustively list the factors without taking the context into consideration.

5. PROPOSED MODEL

This study is based on the comparison of the effectiveness of various requirements elicitation techniques within a rural context, as well as the evaluation of the Task Characteristics Model [3], and the Unified Model of Requirements Elicitation [4]. The purpose is to propose a dynamic model for the elicitation of requirements for rural settings. The strengths of both the Task Characteristics Model and the Unified Model of Requirements Elicitation is the manner in which they recognize that a relationship exists between the problem and solution domains, the stakeholders, the project/organization, and the elicitation process. The main difference is that the Task Characteristics Model recommends the use of specific prompting techniques to interface that relationship, while The Unified Model of Requirements Elicitation recommends a technique selection process to determine how to interface the relationship. This study found that one model is too rigid as it prescribes specific methods for eliciting user requirements, while the other model's rigidity lies in its diagnosis of the factors that determine the requirements elicitation techniques to be applied.

This paper proposes a model that categorizes requirements elicitation techniques into two classes, Conceptual depth and Technical detail, as opposed to previous models which prescribed techniques to improve the requirements elicitation process or explicitly identified factors that contribute to the requirements elicitation technique selection process.

5.1 Rationale for the Categorization

At various stages of the Software Development Lifecycle (SDLC) different types of information are required by the analyst regarding the system to be developed. The depth and detail of information required from the various stakeholders differs throughout this lifecycle. User requirements elicitation is an ongoing process and is not limited to the analysis phase of the SDLC. Similarly, techniques selection should be recognized as an ongoing process in order to sustain the analyst's focus on the application of appropriate techniques throughout the SDLC.

5.1.1 Conceptual Depth

The initial interaction with the user should aim at identifying the problem and creating a case for the project. The proposed model illustrates this point of requirements elicitation as one of maximum conceptual depth and minimal technical detail. This point is characterized by the application of strategies to obtain an in-depth understanding of the various elements surrounding the environment within which the system is intended to be implemented. This includes the knowledge of the prospective users, the existent business processes, socio-political structures, the problems surrounding the targeted community, the skills available amongst the development community, the time

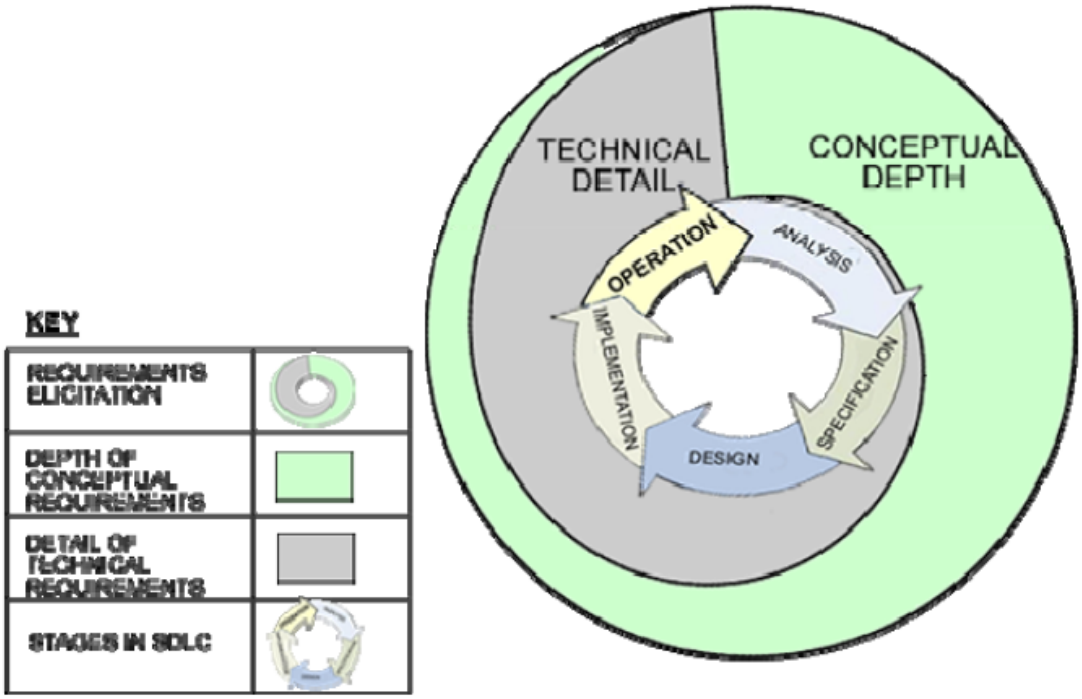


Figure 1: A model for context specific requirements elicitation

constraints, and the resources available. The list of variables affecting the attainment of high conceptual depth requirements is endless, and the depth required varies from project to project. During this stage, the analyst must select appropriate requirements elicitation techniques based on the variables that affect elicitation in the specific project. The prompting techniques (Procedural and Substantive) as defined in the Task Characteristics Model are a useful means of identifying factors affecting technique selection. It is rarely possible or even desirable to allocate strict time deadlines to this stage as it is largely dependent on the response of the stakeholders. If executed effectively, this stage of requirements elicitation should result in the achievement of stakeholder's goals, user satisfaction and the ultimate improvement or solution of the problem. Examples of effective techniques during this stage are interviews, narratives and stories, Joint Application Design (JAD) and focus groups.

5.1.2 Technical Detail

As the project progresses into subsequent stages of the SDLC, the level of conceptual depth required from the elicitation process will decline, while the level of technical detail necessary will increase. At this stage there is an increasing need to translate the requirements into an end product. This stage represents the diminishing need for knowledge regarding the problem domain, and the rising need for definition of the solution domain. This stage is characterized by an increasing dependency on the knowledge, skills and expertise of analysts, decreasing dependency on interaction with stakeholders, predictable time frames for completion, and increasing detail in terms of the technical system requirements. The effective implementation of this stage will determine the overall quality of the end product, the level of usability and the efficiency of the software. This stage makes use of elicitation techniques such as Use-Cases and prototyping.

5.2 Application of the Model in Dwesa

The experience of eliciting user requirements in Dwesa highlighted some of the difficulties of the requirements elicitation process in a rural context. Firstly, one of the most fundamental factors to take into consideration when applying software in Dwesa was the profile of the users. Technique selection required that a thorough knowledge of the prospective users and the problem domain was obtained. This was critical to the success of requirements elicitation and could only be gained through substantial interaction with the user community. This interaction guided the analysts to the set of requirements elicitation tools and techniques that could be applied in the given context, and aided them in overcoming some of the challenges associated with rural ICT implementations. Secondly, effective elicitation was dependent on the analyst's knowledge of and ability to apply the relevant requirements elicitation techniques. Adequate preparation prior to requirements elicitation was necessary. This involved acquainting oneself with the techniques to be applied, and ensuring that the support mechanisms were available during the elicitation process. These included, amongst others, translators and recording devices.

The essence of techniques that yield information high in conceptual depth is that they require intense interaction with the community within which the software will be applied. The result of their application allowed the project group to determine the procedural feasibility, and may ultimately have had the greatest bearing on the long term sustainability of the project. These techniques formed the foundation of the requirements elicitation process as they defined the users' and analyst's problem spaces, and helped to put the project, problem and solution domains into perspective. Requirements elicitation techniques that later defined the technical detail of the application could not have been applied until these elements had been adequately defined.

The Dwesa project was initiated after a needs assessment in 2005, subsequently followed by exploratory interviews, prototypes and ultimately the launch of the E-Commerce platform that currently hosts the products of the entrepreneurs of the area. Although the analysts were not guided by a specific model, their information needs at various points of the project led them to inadvertently elicit the requirements by seeking conceptual depth requirements before pursuing technical detail requirements. The proposed model was developed as a direct result of the analysis of the requirement elicitation activities since the beginning of the project.

As this study is founded on a research dissertation, further empirical work is currently in progress by means of a case study to validate and further refine the model.

6. CONCLUSION

Although the critical evaluation of requirements elicitation techniques may contribute profoundly to the improvement of the requirements elicitation process, each requirements elicitation technique has its own set of advantages and disadvantages. These can vary depending on the context within which the techniques are applied. Thus, there is greater value in defining a methodology to support the application of requirements elicitation techniques.

The Task Characteristics Model [3] is very useful in its definition of prompts to trigger responses in users. Unfortunately, it has only been tested in environments that are distinctly different from most rural ICT experimentation sites. In the case of rural ICT implementations, there is a more pronounced need to refrain from prescribing specific techniques as little is known about the environments.

The Unified Model of Requirements Elicitation [4] is more accommodating to rural software implementation projects as it is founded on the process of selecting techniques based on the knowledge surrounding the problem and the project. One of its shortcomings is that it does not recognise the knowledge surrounding users and stakeholders as a factor affecting the technique selection process. Its main limitation is that it lists the categories of variables that contribute to requirements elicitation technique selection but does not accommodate the existence of any other contributing factors.

The proposed new model captures the strengths of both models discussed [3] and [4]; it recognises the elements that affect requirements elicitation and technique selection. It further addresses their weaknesses by providing a dynamic and accommodating framework for user requirements elicitation that can be adapted according to the stages of the software development lifecycle.

7. REFERENCES

[1] Boehm, B.W. 1981. Software engineering economics. New Jersey: Prentice -Hall.

[2] Brooks, F.P. 1987. No silver bullet: Essence and accidents of software engineering. IEEE Computer, Vol. 20, No. 4, pp. 12-19.

[3] Browne, G.J. and Rogich, M.B. 2001. An empirical investigation of user requirements elicitation: Comparing the effectiveness of prompting techniques. Journal of Management Information Systems, Vol. 17, pp. 223–249.

[4] Davies, A.M. and Hickey, A.M. 2004. A unified model of requirements elicitation. Journal of Management Information Systems, Vol. 20, No. 4, pp. 65–84.

[5] Davis, A.M. 1995. 201 principles of software development. New York: McGraw-Hill.

[6] Fay, D., Timmermans, H. and Palmer, R. 2002. From conflict to negotiation. Human Sciences Research Council Rhodes University. [Online]. Available: http://dwesa.coe.ru.ac.za/wiki/Resources. [Accessed on 18 April 2006]

[7] George, J.F., Hoffer, J.A. and Valacich, J.S. 1998. Modern systems analysis and design, 2nd edition. Boston: Addison and Wesley.

[8] Leffingwell, D. 1997. Calculating the return on investment from more effective requirements management. American Programmer, Vol. 12, No. 4, pp. 13–16.

[9] Rettig, M. 1994. Prototyping for tiny fingers. Communications of the ACM, Vol. 37, No. 4, pp. 21-27.

[10] Schach, S.R. 1998. Classical and object oriented software engineering with UML and C++, 4th edition. New York: McGraw-Hill.

[11] Silver, D. and Wood, J. 1995. Joint application development, 2nd edition. New York: John Wiley.

[12] Wiegers, K.E. 1996. Creating a software engineering culture. New York: Dorset House.