



UNIVERSITY OF CAPE TOWN
IYUNIVESITHI YASEKAPA • UNIVERSITEIT VAN KAAPSTAD

Faculty: Engineering and the Built Environment

Supervisor: Mr Ian Jay

**EFFECTIVE AND EFFICIENT
REQUIREMENT TRACEABILITY IN THE
SOFTWARE DEVELOPMENT AND INFORMATION
TECHNOLOGY
INDUSTRY**

By

Tafadzwa Shereni

SHRTAF001

A minor dissertation

presented to the

Department of Construction Economics and Management

in partial fulfilment of the requirements for the degree

MSc in Project Management

October 2014

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

PLAGIARISM DECLARATION

1. I know that plagiarism is wrong. Plagiarism is using another's work and to pretend that it is one's own.
2. I have used the Harvard system as the convention for citation and referencing. Each significant contribution to, and quotation in, this essay from the work, or works of other people has been attributed and has cited and referenced.
3. This report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as his or her own work.
5. I acknowledge that copying someone else's assignment or essay, or part of it, is wrong, and declare that this is my own work.

Signature

Date

ABSTRACT

Requirements traceability has been identified as a quality factor and a characteristic a system should possess and include as a non-functional requirement. Requirements engineering processes should always include methods and tools of maintaining traces and relationships between requirements and product artefacts. To investigate the extent to which requirements traceability is used in software and information technology projects, a theoretical model of requirements traceability was presented in this research. Five organizations were investigated through semi-structured interviews and their requirements tracing practices were compared with the theoretical model. The extent to which organizations apply requirements traceability practices in their projects differs and as a result they were categorised as inactive, dormant and active users in this research. The advent of agile development methods is one of the major factors affecting requirements traceability practices. Among other recommended areas of further research, there is need for future research to look at how agile development and traditional methods can be implemented together in requirement tracing practices.

DEDICATION

In memory of Agnes Shereni my mother, and Janet Chirasha my grandmother. May the Almighty Lord always keep an eye on you and may your souls rest in eternal peace.

ACKNOWLEDGEMENTS

Firstly, praise be to the Almighty God who gave me this opportunity to take this study and be where I am today. The gift of life and good health He gave through thick and thin shall always be appreciated.

Secondly, I would like to express my gratitude to Mr. Ian Jay my supervisor for the unwavering support, constructive criticisms and guidance throughout this research study. I shall always cherish the wealthy of knowledge and information you provided to me throughout this research study. Your guidance, support and advice have been priceless.

Lastly, I would like to thank my family and friends for the moral support they provided throughout this research study. Special mention goes to Abedinigo Shereni, Nhamo Dhlamini, Makawosi Louis Shereni and their wives. Words cannot express how grateful I am to you all. You are my true heroes.

Table of Contents

PLAGIARISM DECLARATION.....	ii
ABSTRACT.....	iii
DEDICATION.....	iv
ACKNOWLEDGEMENTS.....	v
LIST OF FIGURES.....	xi
LIST OF TABLES.....	xii
ABBREVIATIONS.....	xiii
CHAPTER ONE: INTRODUCTION.....	14
1.1 OVERVIEW.....	14
1.2 BACKGROUND TO THIS STUDY.....	15
1.3 PROBLEM STATEMENT.....	19
1.4 RESEARCH QUESTIONS.....	19
1.5 RESEARCH AIMS.....	19
1.6 RESEARCH OBJECTIVES.....	19
1.7 RESEARCH METHOD.....	20
1.8 LIMITATIONS.....	20
1.9 STRUCTURE OF THE RESEARCH REPORT.....	20
Chapter one: Introduction.....	20
Chapter two: Literature review.....	20
Chapter three: Research methodology.....	21
Chapter four: Analysis of data.....	21
Chapter five: Conclusions and recommendations.....	21
CHAPTER TWO: LITERATURE REVIEW.....	22
2.1 INTRODUCTION.....	22
2.2 REQUIREMENTS.....	22
2.3 REQUIREMENTS ENGINEERING.....	24
2.3.1 Requirement elicitation.....	25
2.3.2 Analysing, modelling and negotiating the requirements.....	25
2.3.3 Communicating/documenting the requirements.....	26
2.3.4 Requirements validation.....	26
2.3.5 Requirements management.....	26
2.4 REQUIREMENTS TRACEABILITY.....	27
2.4.1 Overview of requirements traceability.....	27

2.4.2 Traceability artefacts and links.....	29
2.4.3 Traceability practices.....	32
2.4.3 Factors affecting traceability practice.....	32
2.5 MODELS, TOOLS AND PRACTICES.....	33
2.5.1 Basic techniques.....	33
2.5.2 Automated or commercial tools.....	34
2.5.3 Value-based requirements traceability.....	35
2.5.4 Agile development and traceability.....	36
2.6 PROPOSED MODEL OF REQUIREMENTS TRACEABILITY.....	37
2.7 CONCLUSIONS.....	39
CHAPTER THREE: RESEARCH METHODOLOGY.....	40
3.1 INTRODUCTION.....	40
3.2 WHY QUALITATIVE RESEARCH?.....	40
3.3 WHY MULTIPLE CASE STUDIES?.....	42
3.4 RESEARCH DESIGN.....	44
3.5 DATA COLLECTION.....	45
3.6 DATA ANALYSIS AND PRESENTATION.....	46
3.7 LIMITATIONS.....	47
CHAPTER FOUR: RESEARCH FINDINGS AND DATA ANALYSIS.....	48
4.1 INTRODUCTION.....	48
4.2 DATA ANALYSIS TECHNIQUES.....	48
4.3 PARTICIPATING ORGANISATIONS.....	50
4.3.1 Organisation A.....	50
4.3.2 Organisation B.....	51
4.3.3 Organisation C.....	51
4.3.4 Organisation D.....	51
4.3.5 Organisation E.....	52
4.4 ANALYSIS OF RESEARCH DATA.....	52
4.4.1 Overview of findings.....	52
4.4.2 Requirements and requirements management processes.....	57
4.4.2.1 General overview in theory.....	57
4.4.2.2 Empirical study - common to most organisations.....	57
4.4.2.3 Empirical study – Major differences amongst the organisations.....	57
4.4.3 Requirement traceability.....	58

4.4.3.1 General overview in theory.....	58
4.4.3.2 Empirical study - common to most organisations.....	58
4.4.3.3 Empirical study – Major differences amongst the organisations.....	59
4.4.4 Requirement to product mapping.....	59
4.4.4.1 General overview in theory.....	59
4.4.4.2 Empirical study - common to most organisations.....	59
4.4.4.3 Empirical study – Major differences amongst the organisations.....	60
4.4.5 Requirement change management.....	60
4.4.5.1 General overview in theory.....	60
4.4.5.2 Empirical study - common to most organisations.....	60
4.4.5.3 Empirical study – Major differences amongst the organisations.....	60
4.4.6 Overall analysis of findings.....	60
4.5 CONCLUSIONS.....	63
CHAPTER FIVE: CONCLUSIONS AND RECOMMENDATIONS.....	64
5.1 INTRODUCTION.....	64
5.2 PROBLEM STATEMENT AND RESEARCH QUESTIONS.....	64
5.2.1 Tracing requirements in both directions.....	64
5.2.2 Traceability models and tools.....	65
5.3 FULFILMENT OF AIMS AND RESEARCH OBJECTIVES.....	66
5.4 RECOMMENDATIONS AND AREAS OF FURTHER RESEARCH.....	69
5.5 CONCLUSIONS.....	69
REFERENCES.....	70
Appendix A.....	78
Confidentiality and consent letter.....	78
Appendix B.....	80
Cover letter.....	80
Appendix C.....	81
Signed confidentiality and consent letter.....	81
Appendix D.....	91
Research interview questions.....	91
Requirements.....	91
Requirements Traceability.....	91
Requirements to product mapping.....	92
Change management.....	92

Conclusion.....	92
Appendix E.....	93
Research interview responses.....	93
Organization A.....	93
Requirements and requirements management processes.....	93
Requirements traceability.....	93
Requirements to product mapping.....	94
Requirements change management.....	94
Conclusion.....	95
Organization B.....	96
Requirements and requirements management processes.....	96
Requirements traceability.....	96
Requirements to product mapping.....	97
Requirements change management.....	97
Conclusion.....	98
Organization C.....	99
Requirements and requirements management processes.....	99
Requirements traceability.....	99
Requirements to product mapping.....	100
Requirements change management.....	100
Conclusion.....	100
Organization D.....	101
Requirements and requirements management processes.....	101
Requirements traceability.....	101
Requirements to product mapping.....	102
Requirements change management.....	102
Conclusion.....	102
Organization E.....	103
Requirements and requirements management processes.....	103
Requirements traceability.....	103
Requirements to product mapping.....	104
Requirements change management.....	104
Conclusion.....	104
Appendix F.....	105

Ethics clearance.....105

LIST OF FIGURES

Figure 1: Factors influencing traceability practice.....	33
Figure 2: Proposed model of requirements traceability.....	37
Figure 3: Proposed interaction of model processes.....	38
Figure 4: Components of qualitative research design.....	44

LIST OF TABLES

Table 1: Some important traceability data types and their benefits.....	31
Table 2: Requirements Traceability Tools. INCOSE, 2004.....	34
Table 3: Summary of responses to major aspects in requirements management.....	53
Table 4: Overview of existence of requirements management processes and documents.....	54
Table 5: Organisations value of requirements management processes and tools in relation to project quality and organization performance.....	55
Table 6: Organisation classification in terms of traceability users.....	56
Table 7: Overall comparison of requirements traceability practices.....	61

ABBREVIATIONS

CMMI	Capability Maturity Model Integration
DOD	Department of Defence
INCOSE	International Council on Systems Engineering
ISO	International Standards Organisation
JAD	Joint Application Development
RAD	Rapid Application Development
SEI	Software Engineering Institute

CHAPTER ONE: INTRODUCTION

1.1 OVERVIEW

Quality is one of the major factors used to measure the success of a project (Atkison, 1999). The demand for high quality, technologically driven software products is rising by the day and it has resulted in most projects being delivered late, at a high cost or with less features than initially promised (Carr, 2000). The software and information technology industry is characterised by complexity and unparalleled problems (Asuncion *et al.*, 2007; Brace and Cheutet, 2012; Carr, 2000; Lee *et al.*, 2003). Software projects have continued to fail despite efforts to improve the project management processes. The increase in project complexity and the potential for unforeseen problems increases with the rise in technology (Carr, 2000). In such complex project environments consistent and traceable software requirements are critical elements in the achievement of overall project success (Salem, 2006).

Being able to trace the life of requirements from their origin, through their allocation to components, to the finished product provides a basis for collaboration and control of functionality, quality, and changes (Kirova *et al.*, 2008). Requirement traceability is a measure of the system quality and is recommended by many standards governing the development of systems (Ramesh, 1998). Most software projects go through a software development cycle and use one or more methodologies. At each stage of the development cycle documents and certain artefacts are produced that help project managers to discover potential requirements conflicts, better understand the impact of change requests, and fulfil process quality standards (Heindl and Biffel, 2005). Requirements tracing is emerging as an effective bridge that aligns system evolution with changing stakeholder needs in this ever-changing business and technology environment (Jarke, 1998).

To keep up with the fast pace of business, software projects must handle the frequently changing goals and needs of the customer (Lee *et al.*, 2003). To meet customer expectations in these fast changing business environments, organizations need to effectively and efficiently manage and trace project requirements. The purpose of this research is to investigate the extent to which requirements traceability is used in software and information technology projects and establish its effects on the project quality and success and eventual organizational performance.

1.2 BACKGROUND TO THIS STUDY

The importance and role of requirement traceability in supporting systems development have been long recognized and has been practised since the 1970s (Ramesh, 1998; Kirova *et al.*, 2008). Requirements traceability has been enforced in artificial intelligence (AI) based designs in the 1970s, and has been used in software development in the early 1980s (Jarke, 1998). In systems engineering, traceability has been broadly enforced via national and international standards starting with USA's DOD in the 1980s (Jarke, 1998). Many professional bodies in different industries like medicine, aviation, food processing, software development and space explorations have adopted the use of traceability in order to comply with industry standards and regulations (Kirova *et al.*, 2008; Casey and Caffery, 2013). The software development industry has over the years increasingly placed greater importance on ISO 9001 standard certification and process maturity models such as the SEI maturity model (Lee *et al.*, 2003).

In the late 1990s empirical research showed that systems management practice progressed from the initial simple compliance verification schemata to very sophisticated models and policies for requirements traceability (Domges and Pohl, 1998). However, full capture of all conceivable traces according to these advanced models was neither desirable nor feasible because the process was time consuming, costly, error-prone, and labour intensive (Domges and Pohl, 1998; Hayes *et al.*, 2003; Hayes *et al.*, 2007).

For the past two decades, many researchers have contributed to the topic of traceability providing solutions in the form of models, methods, tools, and a better understanding of traceability needs and challenges (Nair, de la Vara and Sen, 2013). The benefits of requirements traceability are widely accepted nowadays and sophisticated tool support is available to record, manage, and retrieve trace information (Egyed and Grünbacher, 2002). Many researchers and practitioners in existing literature have come with different models and tools that aid in requirement traceability practice. Some of the earliest known models are; the evolution support environment (ESE) system; PRISM model of changes; The NATURE project; and the DOD model (Wieringa, 1995).

Some of the earliest techniques that have been used for providing requirement traceability includes: cross referencing schemes, key phrase dependencies, templates, requirement traceability matrices, matrix sequences, hypertext, integration documents, assumption-based truth maintenance networks, and constraint networks (Gotel and Finkelstein, 1994). Automated support tools such as general-purpose tools, special-purpose tools, workbenches and CASE tools are tools that are used to provide

traceability (Wieringa, 1995; Gotel and Finkelstein, 1994).

Currently there are several modern commercial tools and environments for requirement traceability as identified by INCOSE. Amongst them include: DOORS by Quality Systems and Software (QSS), icCONCEPT by Integrated Chipware, XTie-RT by Teledyne Brown Engineering, DOORSrequireIT by Quality Systems and Software (QSS), and Rational RequisitePro by Rational Software (International Council of System Engineering [INCOSE], 2004). Despite all these models and tools, the lack of effective software traceability continues to be a perennial problem in industry projects (Asuncion *et al.*, 2007).

There are other emerging and proposed tools and models of traceability that are indicated in the literature. One of the proposed models is a value-based requirements tracing process which identifies traces based on prioritized requirements and thus identify traces that are more important and valuable than others (Heindl and Biffel, 2005). The approach provides a technical model and an economic model for requirements tracing, depending on criteria like number of requirements, value of requirements, risk of requirements, number of artefacts, number of traces, precision of traces, size of artefacts, cost/effort of trace identification and maintenance, and value of traces (Heindl and Biffel, 2005). Another proposed model is a concept of an end-to-end traceability that extends throughout the entire life of a development project (Asuncion *et al.*, 2007). Existing tools failed to solve the traceability problem due to tool rigidity, narrow focus, and lack of interoperability with other tools (Asuncion *et al.*, 2007). Another model is the event-based traceability model for managing change which seek to provide support timely updates for artefacts and related links impacted by change (Cleland-Huang *et al.*, 2003).

The problem of requirement traceability is quite broad and the first issue with it is that practitioners do not use the tools, techniques and models mentioned above to control the information about the requirements they want to trace (Gotel and Finkelstein, 1995). The use of informal traceability methods, failure in the cooperation between people responsible for coordinating traceability, difficulty in obtaining necessary information in order to support the traceability process, and lack of training of personnel in traceability practices are contributing factors to the problem of traceability (Salem, 2006). Two significant findings from this problem were that:

1. *There was lack of agreement regarding the quantity and type of information that practitioners wanted to trace about requirements,*
2. *There was extreme importance attached to personal contact and informal communication*

(Gotel and Finkelstein, 1995).

The second issue with traceability is the generation and maintenance of traceability links. The problems with the tools and models of traceability in terms of traceability links were summarized by Hayes *et al.*, (2003), as follows:

1. *They require performance of interactive searches for potential linking requirements or design elements,*
2. *They require the assignment of keywords to all the elements in both document levels prior to tracing,*
3. *They return many potential or candidate links that are not correct,*
4. *They fail to return correct links, and*
5. *They do not provide support for easily retracing new versions of documents. (Hayes et al., 2003).*

At the heart of the traceability problem is “*the sheer number of artefacts produced in a project, the differing levels of formality and specificity between various artefact types, and the complex interrelationships between artefacts*” Asuncion *et al.*, (2007). While traceability links help to scope the possible impact of change, they do not support automated reasoning about change, because the links carry little semantic information (Nuseibeh and Easterbrook, 2000). Requirement traces are in a constant state of flux since they may change whenever requirements or other development artefacts change (Egyed and Grunbacher, 2002). Software development standards are unclear on what information, artefact properties and relationships should be recorded to improve the efficiency of the development process (Arkley and Riddle, 2005).

Software development environments are characterised by change (Zowghi and Nurmuliani, 2002; Kirova *et al.*, 2008). Requirements constantly change and evolve during the project development life cycle and it is for this reason that software projects have become too complex (Zowghi and Nurmuliani, 2002; Salem, 2006). The level of complexity of customer requirements has also increased due to increase in product complexity and distributed development (Brace and Cheutet, 2012). It is therefore important that all documents produced during the project life-cycle be linked and each lower level requirement should be traceable to higher level requirements (Carr, 2000). Components created at various stages of the development process, include hardware, software, code modules, designs, humanware, manuals, test cases and results, policies and procedures should be

linked and traceable to the requirements (Ramesh and Edwards, 1993).

Requirements traceability has been identified in the literature as a quality factor and as a characteristic a system should possess and include as a non-functional requirement (Ramesh and Jarke, 2001). It makes knowledge about the designed system independent from people and as such can improve system quality (Heindl and Biffel, 2005). Important goals of requirements traceability are to facilitate communication, to support integration of changes, to preserve design knowledge, to assure quality, and to prevent misunderstandings (Egyed and Grünbacher, 2002). Neglecting traceability or capturing insufficient and/or unstructured traces leads to a decrease in system quality, causes revisions, and thus, increases project costs and time (Domges and Pohl, 1998). The generation and maintenance of relationships and links can provide a basis for more effective system quality assurance, management of changes, and software maintenance (Spanoudakis et al, 2004).

Traceability is necessary to drive and verify completeness and consistency, to manage complexity and churn caused by a variety of factors throughout the life-cycle of a system, and to allow for project monitoring and control (Egyed and Grünbacher 2002). It aids in system comprehension, impact analysis, system debugging, and communication between the development team and stakeholders (Asuncion *et al.*, 2007). It also provides a means of establishing and clarifying the contractual requirements, allows the development engineers to control their problem space (requirements and design) and provides means to demonstrate the validity of their product (Arkley and Riddle, 2005). It directly alleviates the problems caused by poor communication and coordination because it automates the role of change notification (Cleland-Huang *et al.*, 2003).

Mader and Egyed (2012) conducted an assessment to investigate whether the use of requirements traceability can significantly support development tasks to eventually justify its costs. They conducted a controlled experiment with 52 subjects performing real maintenance tasks on two third-party development projects and they found out that:

“subjects with traceability performed on average 21% faster on a task and they created on average 60% more correct solutions” (Mader and Egyed 2012).

A comprehensive scheme for maintaining traceability, especially for complex, real-time systems, requires that all system components, not just software, created at various stages of the development process be linked to the requirements (Ramesh *et al.*, 1995). A primary concern in the development of large-scale, real-time, complex, computer-intensive systems is ensuring that the performance of

system meets the specified requirements (Ramesh and Edwards, 1993). Requirement traceability has been around for decades but still projects are failing and there is a no clear agreement on generic models and tools that can be used in improving the quality of software through tracking and tracing of important and valuable requirements (Egyed and Grunbacher, 2002; Asuncion *et al.*, 2007).

1.3 PROBLEM STATEMENT

The problem to be looked at in this research study is defined as follows:

In software development and information technology projects, requirements constantly change and become more complex, these changes are not being adequately traced throughout the project life cycle resulting in project delays, high project costs, poor quality products and in most cases premature project termination.

1.4 RESEARCH QUESTIONS

The two research questions to be addressed in this research study are:

1. Are software and information technology companies tracing project requirements in both forward and backward directions?
2. Are there recognized requirement traceability models and tools that have been adopted by software and information technology companies in Cape Town, how effective and efficient are they and how are they helping in improving quality and project deliverance?

1.5 RESEARCH AIMS

The aim of this research study is to determine how well the idea of tracing requirements is used in the software and information technology industry and what role it plays in producing a quality and successful project.

1.6 RESEARCH OBJECTIVES

The research objectives to be achieved in this research are as follows:

1. Determine how requirements are collected and documented in software projects environments.
2. Determine how requirements are being traced in software projects.
3. Identify factors that are affecting requirement traceability in software projects.

4. Identify important aspects and issues that should be captured and addressed by requirement traceability.
5. Identify traceability tools and models that are being used in the software and IT companies in Cape Town.

1.7 RESEARCH METHOD

The questions, aims and objectives of this research study will be answered, addressed and achieved by focusing on understanding requirements traceability issues in software development and IT organizations using a qualitative research approach. Multiple case studies will be adopted as strategy of inquiry. Semi-structured interviews will be used as a method to collect data and these interviews will be conducted with selected software development and IT organisations in Cape Town.

1.8 LIMITATIONS

The limitations in this research study are:

- The major limitation in this study is the sample size of software and IT companies that are willing to provide information on requirements management and traceability.
- The study will be restricted to software and information technology companies based in Cape Town due to geographical and economic factors.
- Software companies are reluctant to provide information and details on how they run projects due to patent and legal issues in software products.

1.9 STRUCTURE OF THE RESEARCH REPORT

This research study report will be structured as follows:

Chapter one: Introduction

Chapter one provides the introduction and brief background of the study that is explained, reported and explored in the rest of this document. It presents the stated problem area of the research and the questions that the research will seek to obtain answers for. The research aims, proposition, objectives, methodology and limitations are stated and described.

Chapter two: Literature review

This chapter will explore and review the literature that exist in relation to project requirements

traceability. The chapter will review and describe requirements and requirement engineering in details. It will then describe in details the practice of requirements traceability. Old and current tools, techniques, models and frameworks used in traceability practices will be reviewed and explored to establish an understanding of how different industries are handling the issue of tracing requirements.

Chapter three: Research methodology

This chapter explains the qualitative methodology that is going to be used in this research study.

Chapter four: Analysis of data

The data analysis chapter is what it is, the analysis of the gathered data and provide a thorough discussion on the findings.

Chapter five: Conclusions and recommendations

This chapter concludes the research study by proving recommendations and suggestions for future research.

CHAPTER TWO: LITERATURE REVIEW

2.1 INTRODUCTION

Medical devices, telecommunication systems, space exploration machines, transportation systems, mega structures and safety-critical systems are dependent on high quality and reliable software (Casey and Caffery, 2013; Iqbal *et al.*, 2011). Software development endeavours in today's business and commercial world are characterized by multiple dependencies, compliance to standards and government regulations, rapid technology innovation, team diversity, and dynamic changes in requirements, business environments, and markets (Kirova *et al.*, 2008). In order for organizations to cope with these complex environments, unexpected situations or changes as well as the need for success it is essential to have clear linkages and traceability from requirements throughout the different stages of the software development and maintenance life cycle (Casey and Caffery, 2013).

In many projects several stakeholders are involved and they provide different requirements, components and services. Supporting the collaboration, managing the development, monitoring the project, and controlling the quality of the product in such distributed settings require methods and tools that, among other things, allow for open but secure information exchange and promote shared understanding of artefact and component dependencies (Kirova *et al.*, 2008). The ability to trace requirements artefacts through the stages of specification, architecture, design, implementation, and testing is a significant factor in assuring a quality software implementation (Leffingwell and Widrig, 2002)

Projects often miss or do not address requirements and/or the impact of change, as a result even small changes to a system can create significant safety and reliability problems (Leffingwell and Widrig, 2002). Consistent and traceable software requirements are critical elements in today's complex software projects (Salem, 2006).

2.2 REQUIREMENTS

Requirements are the descriptions of properties, attributes, services, functions, and/or behaviours needed in a product to accomplish the goals and purposes of the system (Carr, 2000). They specify what the system must do and must incorporate its objectives, life cycle, operational modes, constraints, and interfaces with other systems (Nicholas and Steyn, 2012). They consists of quantified and documented needs and expectations of the project sponsor, customer and other stakeholders (PMI, 2008). A requirement should fulfil a current or future need by stakeholders

(Heindl and Biffel, 2005). The overall agreement within literature is that requirements must address and reflect the needs of all the stakeholders who benefits or are impacted by the system (PMI, 2008; Asghar and Umar, 2010; Nicholas and Steyn, 2012). Requirements can include business requirements, user requirements, functional requirements, non-functional requirements, delivery requirements, performance requirements, and process requirements (PMI, 2008; Salem, 2006).

Requirements are generated from the way people work and interact with the system and its environment or its application domain (Asghar and Umar, 2010). They reflect different interests and perspectives of the different stakeholders, which may include characteristics like durability, operating costs, performance, reliability, safety and capacity (Nicholas and Steyn, 2012). These aspects help to identify what would be the systems requirements and through requirements analysis they become part of the system specification document.

System requirements explain the detailed description of what software is supposed to do (Asghar and Umar, 2010). They provide an overview of the system or solution approach, the principle functions, system architecture and resulting end product (Nicholas and Steyn, 2012). They simple state what the system must do to satisfy the user requirements (Carr, 2000; Nicholas and Steyn, 2012). These requirements are classified as functional or non-functional, and may take on different forms (Carr, 2000; Asghar and Umar, 2010; Nicholas and Steyn, 2012).

Functional requirements are those things that the system must do, they specify the functions that the new system must able to do perform to meet the user requirements (Carr, 2000; Nicholas and Steyn, 2012). Non-functional requirements are mostly constraints on the system design (Carr, 2000). Each functional requirement does not exist in isolation, it is tied to its own constraints within the entire system or to other requirements (Kotonya and Sommerville, 1996). System characteristics like reliability, usability, robustness, maintainability, compatibility and expandability are all constraints on the system (Nicholas and Steyn, 2012).

Identifying a problem's context, locating the customer's requirements within that context and delivering a specification that meets customer needs within that context is usually an arduous task that needs proper management (Verner *et al*, 2007). This is where the ideas of requirements engineering are called into action. In the next section we are going to look at how requirements engineering help in managing requirements.

2.3 REQUIREMENTS ENGINEERING

Requirements engineering is the process of discovering the degree to which a software system meets the purpose for which it was intended, by identifying stakeholders and their needs, and documenting these in a form that is amenable to analysis, communication, and subsequent implementation (Nuseibeh and Easterbrook, 2000). In essence, it attempts to communicate the ideas and needs of the stakeholders to the engineers and developers who build the software system or product (Lee *et al.*, 2003). The premise behind it is to document stakeholder requirements, design rationale, and traceability to improve the quality of software development and the efficiency of software upgrade and maintenance (Lee *et al.*, 2003). Software requirements engineering involves managing a variety of knowledge about the various activities related to the software process (Neto and Morais, 2013).

Documents created and maintained during requirement engineering, project development and throughout the life cycle of the project should be traceable, with links well established (Wieringa, 1995). Similarly, a requirement should be traceable throughout the life of a project with lowest level requirement traceable to higher level requirements (Carr, 2000).

Requirements engineering should look at three dimensions:

1. *Managing the convergence of stakeholder interests toward agreement on key system goals and constraints (Jarke, 1998).*
2. *Achieving a sufficient shared understanding of the issues involved in realizing the system vision, such as its functionality, non-functional properties, intended and unintended side effects (Jarke, 1998).*
3. *Documenting this understanding in adequate representation formats, for human information sharing as well as for computerized system development (Jarke, 1998).*

All projects go through a lot of changes in their lifetime and in order to manage these changes effectively, organizations need to embrace the ideas of requirement engineering. Although requirements change during project development and evolve when a system is operational, requirements engineering is often regarded as a front-end activity in the software systems development process (Nuseibeh and Easterbrook, 2000). It consists of several activities which include; requirements elicitation, modelling and analyzing the requirements, communicating stakeholder needs, transforming customer requirements into derived requirements, agreeing or negotiating the requirements, allocating requirements to hardware, software and interface elements,

requirements verification, requirements validation and maintaining requirements (Carr, 2000; Nuseibeh and Easterbrook, 2000 ; Bahill and Henderson, 2005; Asghar and Umar, 2010).

The main goal of requirement engineering is to meet the degree of end user's satisfaction in minimum cost and time (Asghar and Umar, 2010). Most errors in software projects occur in the requirements phase (Salem, 2006), hence requirements engineering has contributed immensely in producing quality software. The quality of the software are contingent to requirement elicitation, requirement analysis and requirement management (Asghar and Umar, 2010), as such we are going to briefly explain the various requirements engineering activities in the next section.

2.3.1 Requirement elicitation

The first step in the requirements engineering process is requirements elicitation, which involves collecting all the requirements for a system or project from all stakeholders (Nuseibeh and Easterbrook, 2000). This is the stage where the problems in existing software systems are investigated and identified (Asghar and Umar, 2010). The main goals of requirements elicitation include; identification of system boundaries through understanding of the problem that need to be solved, identification of all stakeholders i.e. individuals and organizations that will be affected by success or failure of the system or project (Nuseibeh and Easterbrook, 2000; Asghar and Umar, 2010). Requirements elicitation techniques and practices include use of questionnaires, interviews, surveys, workshops (RAD and JAD sessions), brainstorming, prototyping and cognitive techniques (Nuseibeh and Easterbrook, 2000).

2.3.2 Analysing, modelling and negotiating the requirements

Requirements analysis checks requirements for necessity or the need, consistency, completeness, and feasibility (Paetsch *et al*, 2003). Joint application development (JAD sessions), prioritization and modelling are some of the techniques that are used to analyse the requirements (Paetsch *et al*, 2003). Modelling is the construction of abstract descriptions that are amenable to interpretation (Nuseibeh and Easterbrook, 2000). It deals with understanding the structure of the organization where development takes place or in which a system will operate (Nuseibeh and Easterbrook, 2000). Modelling and analyses tries to understand the business rules that affect the organization's operation; the goals, tasks and responsibilities of its constituent members; and the data that it needs, generates and manipulates (Nuseibeh and Easterbrook, 2000).

As requirements are elicited and modelled, maintaining agreement with all stakeholders can be a

problem, especially where stakeholders have divergent goals (Nuseibeh and Easterbrook, 2000). Briggs and Gruenbacher (2002) argues that in requirements negotiation the complexities are highly dynamic and as such the process should engage success-critical stakeholders. A success-critical stakeholder is any individual whose interests must be accommodated in order for the project to succeed (Briggs and Gruenbacher, 2002). Requirements emerge in a highly collaborative, interactive, and interdisciplinary negotiation process that involves heterogeneous stakeholders (Boehm, Grunbacher, and Briggs, 2001).

2.3.3 Communicating/documenting the requirements

Requirements documentation is concerned with facilitating effective communication of the requirements among different stakeholders and developers (Nuseibeh and Easterbrook, 2000; Paetsch *et al*, 2003). These documents play a major role in communicating the requirements to stakeholders. They must contain requirements which are easily read, analysed, written, and validated (Nuseibeh and Easterbrook, 2000). A good requirements document is unambiguous, complete, correct, understandable, consistent, concise, and feasible (Paetsch *et al*, 2003). Requirements traceability is a major factor that determines how easy it is to read, navigate, query and change requirements documentation (Nuseibeh and Easterbrook, 2000).

2.3.4 Requirements validation

Requirements validation is the process of establishing that the requirements and models elicited provide an accurate account of stakeholder requirements (Nuseibeh and Easterbrook, 2000). The process help in certifying the requirements as an acceptable description of the system to be implemented (Paetsch *et al*, 2003). Validating requirements means ensuring that:

1. The set of requirements is correct, complete, and consistent,
2. A model that satisfies the requirements can be created, and
3. A real-world solution can be built and tested to prove that it satisfies the requirements (Bahill and Henderson, 2005)

2.3.5 Requirements management

The goal of requirements management is to capture, store, disseminate, and manage information, and it includes all activities concerned with change and version control, requirements tracing, and requirements status tracking (Paetsch *et al*, 2003). Requirements should be written in a way that is readable and traceable, in order to manage their evolution over time (Nuseibeh and Easterbrook, 2000). Successful software systems always evolve as the environment in which these systems

operate changes and stakeholder requirements change (Nuseibeh and Easterbrook, 2000). This involves providing techniques and tools for configuration management and version control, and exploiting traceability links to monitor and control the impact of changes in different parts of the documentation (Nuseibeh and Easterbrook, 2000).

One aspect that has been spelt out in requirements management is the ability to trace and link requirements. Requirements traceability lies at the heart of requirements management practice and it provide a rationale for requirements and is the basis for tools that analyse the consequences and impact of change (Nuseibeh and Easterbrook, 2000). In the next section requirement traceability is described in more detail.

2.4 REQUIREMENTS TRACEABILITY

2.4.1 Overview of requirements traceability

Requirements traceability is defined as the ability to describe and follow the life of a requirement, in both a forward and backward direction (Gotel and Finkelstein, 1994; Wieringa, 1995; Domges and Pohl, 1998; Jarke, 1998). According to the Institute of Electrical and Electronics Engineers (IEEE) traceability is “*the degree to which a relationship can be established between two or more work products of the development process, especially work products having a predecessor-successor or master-subordinate relationship to one another.*” (Kirova *et al.*, 2008). What can be clearly deduced from the two definitions above is that there should be bidirectional traceability among the requirements and other work products (Kirova *et al.*, 2008). A requirement should be traceable from its origins, through its development and specification, to its subsequent deployment and use, and through periods of ongoing refinement and iteration in any of these phases (Gotel and Finkelstein, 1994).

As describe earlier, the practice of requirements traceability in system development has been long recognized since the 1970s and has went through many changes over the past two decades (Ramesh, 1998; Kirova *et al.*, 2008; Nair, de la Vara and Sen, 2013). The importance of traceability has been recognized by many regulatory bodies and organizations in various industry sectors and it has been subsequently incorporated into various standards and guidelines (Mader *et al.*, 2013).

The first well known analysis of requirements traceability issue was conducted by Gotel and Finkelstein in 1994. They conducted an empirical research involving over 100 practitioners and

they found out that they was no common understanding amongst the practitioners on what constitutes requirements traceability (Gotel and Finkelstein, 1994).

There are several benefits of requirements traceability to the product organization and to the end customer highlighted in the literature. Improved product quality, effective response to changes, controlled requirements and product churn, and a resulting reduction in the product implementation interval (Kirova *et al.*, 2008), are some of the benefits of traceability.

Software projects are characterised by rapidly changing development and operational environments. Requirements tracing is an effective bridge that aligns system evolution with changing stakeholder needs and it helps uncover unexpected problems and innovative opportunities, and lays the groundwork for corporate knowledge management (Jarke, 1998). It provides relationships between requirements, design, and implementation of a system in order to manage changes to a system (Paetsch *et al.*, 2003). Providing requirements traceability in requirements documentation is a means of achieving integrity and completeness of that documentation, and has an important role to play in managing change (Nuseibeh and Easterbrook, 2000). It ensures that project stakeholders have complete and consistent information about the system or product being built (Kirova *et al.*, 2008).

Requirements traceability has been identified as a quality factor and characteristic a system should possess and include as a non-functional requirement (Ramesh and Jarke, 2001). For organizations and project practitioners to reach higher project maturity levels and produce better quality products they must effectively trace requirements (Rempel *et al.*, 2013). CMMI has been using requirements traceability as a benchmark for quality standards and to be appraised at CMMI maturity levels, product organizations must maintain bidirectional traceability among the requirements and work products (Heindl and Biffel, 2005; Kirova *et al.*, 2008). It has been recognized as a significant contributor to efficient software and system quality (Hayes *et al.*, 2007).

Requirement traceability ensures that “*the right product is built*” and that “*the product is built right*” (Kirova *et al.*, 2008). It ensures that the right product is built by:

1. Identifying the need addressed by a requirement and hence establishing that a requirement is necessary,
2. Identifying how to interpret a requirement and what design decisions affect the implementation of a requirement,
3. Identifying that all requirements are allocated to design, and

4. Identifying that all design elements are necessary. (Kirova et al., 2008)

It also ensures that the product is built right by:

1. Identifying what tests are used to verify a requirement and hence allowing a check into whether a test is suitable and necessary.
2. Checking that all requirements are covered by test cases, what are the outcomes of executing the test, and that a risk is identified and tracked if some requirements cannot be tested.
3. Helping to identify if the implementation is in compliance with the requirements. (Kirova et al., 2008)

Requirement traceability lies at the heart of requirements management practice in that it can provide a rationale for requirements and is the basis for tools that analyse the consequences and impact of change (Nuseibeh and Easterbrook, 2000).

Although requirements traceability has been around for decades, there still exists problems within its practice. The major issues raised in the literature are that the processes involved are time consuming, costly, error-prone, and labour intensive (Domges and Pohl, 1998; Hayes *et al.*, 2003; Hayes *et al.*, 2007; Mader *et al.*, 2013). Creating and maintaining trace links can be an arduous, error-prone, and costly process that can have a significant effect on the overall costs and time-to-market for a product (Mader *et al.*, 2013).

One of the problems is that tracing is typically not an explicit systematic process, but occurs on an *ad hoc* basis with considerable hidden tracing-related quality costs (Heindl and Biffel, 2005). Identifying and maintaining trace dependencies leads to additional effort that can get prohibitively expensive with an increasing number of requirements and increasing tracing precision (Heindl and Biffel, 2005). Another problem is that the way traceability tools present the data is often difficult to navigate and there is uncertainty about how the traceability information is employed (Arkley and Riddle, 2005).

2.4.2 Traceability artefacts and links

Requirements traceability is concerned with relating requirements specifications with other artefacts created in the development life-cycle of a software system (Spanoudakis *et al.*, 2004). Traceability relations are importance in several ways:

1. They assist the process of verifying that a system meets its requirements (Spanoudakis *et al.*, 2004).

2. They are used to establish the impact of changes in the requirement specification of a system on other artefacts in its documentation and vice versa (Spanoudakis *et al.*, 2004).
3. They are used to understand the evolution of an artefact (Spanoudakis *et al.*, 2004).
4. They are used to understand the rationale underpinning certain design and implementation aspects of a system ((Spanoudakis *et al.*, 2004).

Software configuration management techniques are commonly used to control the evolution of software systems through providing version control, configuration identification, and support for accounting activities related to change management (Cleland-Huang *et al.*, 2003). Software configuration management is used to establish links and traces whenever changes are done to the project. Traceability links define the relationships that exist between the various artefacts of the software engineering process (Cleland-Huang *et al.*, 2003). Requirements tracing consists of document parsing, candidate link generation, candidate link evaluation, and traceability analysis (Hayes *et al.*, 2004).

Requirement traceability can be divided into two:

1. **pre-RS (pre-requirement specification) traceability** which refers to those aspects of a requirement's life prior to inclusion in the requirement specification and,
2. **post-RS (post-requirement specification) traceability** which refers to those aspects of a requirement's life that result from inclusion in the requirement specification

(Gotel and Finkelstein, 1994).

Four kinds of traceability links are explain as follows:

1. **Forward from requirement:** Responsibility for requirements achievement must be assigned to system components, such that accountability is established and the impact of requirements change can be evaluated (Jarke, 1998).
2. **Backward to requirements:** Compliance of the system with requirements must be verified, and gold-plating (designs for which no requirements exist) must be avoided (Jarke, 1998).
3. **Forward to requirements:** Changes in stakeholder needs, as well as in technical assumptions, may require a radical reassessment of requirements relevance (Jarke, 1998).
4. **Backward from requirements:** The contribution structures underlying requirements are crucial in validating requirements, especially in highly political settings (Jarke, 1998).

In order to effectively capture the data to be traced and produce effective links between artefacts the

project manager need to define the data types that can be captured (Domges and Pohl, 1998). Table 1, below show the traceability data types and their benefits.

Traceability data types	Typical benefits
Bi-directional links, between customer requirements, derived requirements, and software components.	The interrelations support the (automated) <ol style="list-style-type: none"> 1. <i>Validation</i> the system functionality meets the customer requirements and that no superfluous functionality has been implemented. 2. <i>Impact analysis</i> upon changing customer requirements. The links allow to determine which derived requirements, design, and implementation parts might be affected.
contribution structure, stakeholders, roles, policies	Exploiting established contribution structures <ol style="list-style-type: none"> 1. <i>Improve communication and cooperation</i> among teams. In the case of a change request, the stakeholders to be involved and/or informed can be determined. 2. Assure the <i>contribution</i> of (leaving) individuals are passed on and thus are not lost.
design rationale, design decisions, alternatives, underlying assumptions	Capturing design rationale improves <ol style="list-style-type: none"> 1. The <i>understanding</i> of the system by the customer and thus the system acceptance. The system behaviour can be justified using the recorded design decisions and the recorded assumptions about expected operating conditions for the system. 2. <i>Change management</i> by reducing the changes of neglecting important considerations during change integration, since previously rejected solutions and the reasons for their rejection are accessible.
process data, tasks performed, consumed resources, quality measures	Process data empowers the project manager to improve <ol style="list-style-type: none"> 1. <i>Software project planning</i>, planning can be based on past histories of similar projects and thus results in more realistic cost and schedule estimates. 2. <i>Software project control</i>, project progress can be measured based on the data collected through binary reporting and/or tracking techniques.

Table 1: Some important traceability data types and their benefits. (Domges and Pohl, 1998).

2.4.3 Traceability practices

Mader *et al.*, 2013 identified six practices for strategic traceability in safety-critical projects which are also applicable to software projects. The following are the six practices:

1. **Plan traceability:** It is important for project managers to strategically plan traceability in a project's early phases and document it using suitable traceability models. The model should be able to trace all artefacts and links associated with them (Mader *et al.*, 2013).
2. **Offer traceability tool support:** Tracing should be supported by automated commercial tools that provides features for establishing, maintaining, and navigating trace links and has the ability to display trace information in formats such as matrices or trace slices (Mader *et al.*, 2013).
3. **Create traceability incrementally:** Instrumenting the environment with tracing tools empowers knowledgeable project stakeholders to create trace links incrementally within the context of their daily work. In most cases, the task of creating, evaluating, and approving traceability links is done very late in the project and is carried by inexperienced individuals which results in links which are often incomplete and inaccurate and aren't available throughout the project to support development (Mader *et al.*, 2013).
4. **Model traceability queries:** Traceability queries cover basic life-cycle activities such as finding all requirements associated with currently failed test cases or listing all mitigating requirements associated with a given hazard. Queries act as filters to that eliminate unwanted artefacts and links (Mader *et al.*, 2013).
5. **Visualize trace slices:** Instead of presenting traceability material in the form of trace matrices, generate trace slice visualizations in which the hazard is the root node and all direct and indirectly traced artefacts that contribute to mitigating the hazard are shown as a tree (Mader *et al.*, 2013).
6. **Evaluate traces continually:** Visually or graphically provide the tracing progress for a project to all stakeholders so that they can easily create appropriate links for the project (Mader *et al.*, 2013).

2.4.3 Factors affecting traceability practice

Ramesh 1998, identified environmental context, organizational context and system development context as factors that influence the practice of traceability. In addition to the factors he also identified two groups of requirement traceability users and these are low-end and high-end traceability users. Low-end users view requirements traceability as a mandate from the project sponsor, while high-end users view it as an important component of quality system engineering

process (Ramesh, 1998). Figure 1, below shows the three factors that influence traceability practice i.e. environmental context, organizational context and system development context (Ramesh, 1998).

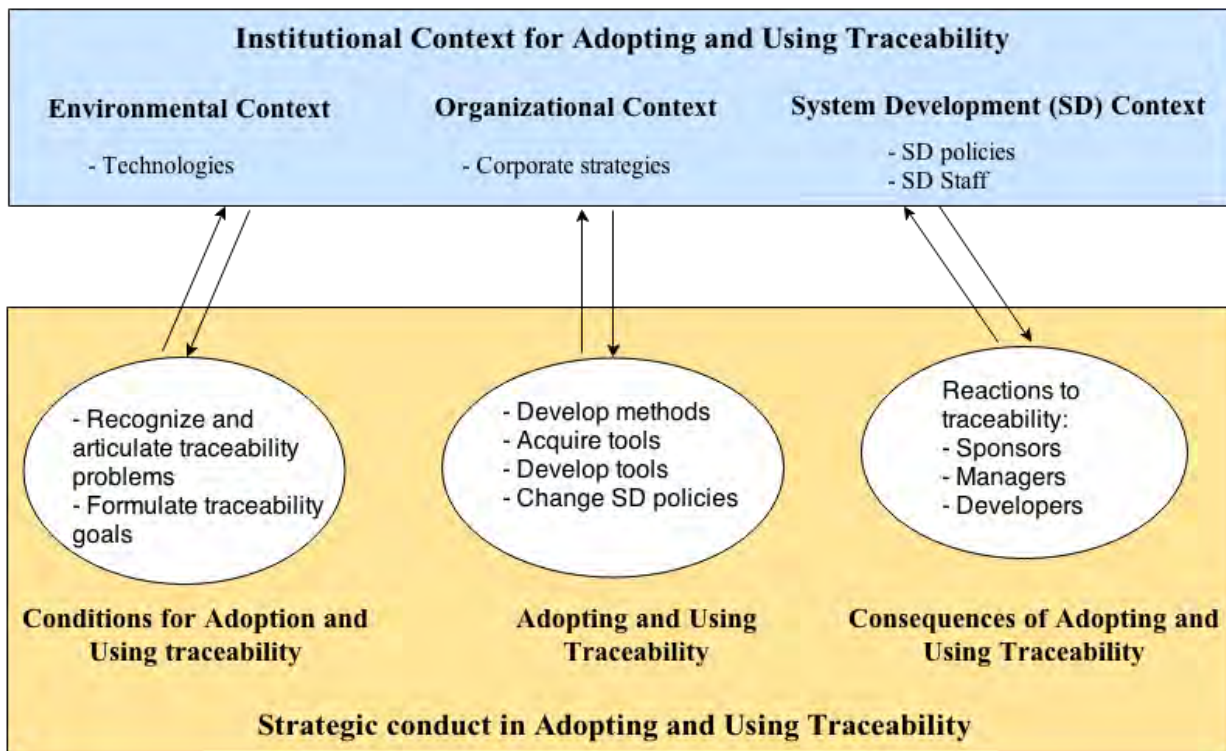


Figure 1: Factors influencing traceability practice.

2.5 MODELS, TOOLS AND PRACTICES

There are several models, tools and techniques identified in literature that have been used by organizations in the implementation of requirement traceability. This section explains briefly all the schools of thought that have emerged in the area of requirements traceability. The earliest classification of models, tools, techniques and practices that support requirements traceability are found in Gotel and Finkelstein, 1994, and these were classified as; basic techniques and automated tools.

2.5.1 Basic techniques

Various basic techniques that were identified include, requirement traceability matrices, cross referencing schemes, matrix sequences, templates, keyphrase dependencies and integration documents (Gotel and Finkelstein, 1994). The techniques differ in the quantity and diversity of information they can trace between artefacts, in the number of interconnections they can control between information, and in the extent to which they can maintain requirement tracing when faced with on-going changes to requirements (Gotel and Finkelstein, 1994).

2.5.2 Automated or commercial tools

General-purpose tools, workbenches, special-purpose tools and environments are identified classifications of automated tools (Gotel and Finkelstein, 1994). General-purpose tools include hypertext editors, word processors, spreadsheets (MS Excel) and database systems. Special-purpose tools support dedicated activities related to requirements engineering and some achieve restricted requirement traceability (Gotel and Finkelstein, 1994). Workbenches are a collection of general-purpose tools and special-purpose tools (Gotel and Finkelstein, 1994).

Table 2, below shows several modern commercial tools and environments for requirement traceability as identified by INCOSE, 2004.

Tool Name	Vendor	Description
Caliber-RM	Technology Builders, Inc (TBI)	Requirements traceability tool
DOORS	Telelogic	Requirements traceability tool
DOORS/ERS	Telelogic	Requirements traceability tool
icCONCEPT	Integrated Chipware	Requirements traceability tool
RDT	IGATECH Systems Pty Limited	Requirements traceability tool
RequisitePro	Rational Software	Requirements traceability tool
DOORSrequireIT	Telelogic	Requirements trace tool that is integrated with Microsoft Word. Data can be merged with DOORS databases
XTie-RT	Teledyne Brown Engineering	Requirements traceability tool
Tracer	RBD, Inc.	Requirements traceability tool
Systems Engineer	Blue Spruce	Requirements traceability tool
GMARC	Computer Systems Architects (CSA)	Generic Modelling Approach to Requirements Capture (GMARC). Toolset will also generate quality metrics for a specification enabling formal proof that use of the GMARC has improved the requirement set.

CORE	Vitech Corporation	Full life-cycle systems engineering CASE tool. It supports the systems engineering paradigm from the earliest days of concept development and proposal development, requirements management, behaviour modelling, system design and verification process.
------	--------------------	---

Table 2: Requirements Traceability Tools. INCOSE, 2004.

2.5.3 Value-based requirements traceability.

One important contributing insight in the practice of traceability was given by Heindl and Biffel, (2005). They carried out a case study on what they called **value-based requirements tracing** (VBRT) that systematically supports project managers in tailoring requirements tracing precision and effort based on stakeholder value, requirements risk/volatility, and tracing costs (Heindl and Biffel, 2005). The goal of value-based requirements tracing is to identify which requirements are more important and valuable. Relative importance would be decided based on a process of prioritising requirements. The VBRT approach provides a technical model and an economic model for requirements tracing, depending on criteria like number of requirements, value of requirements, risk of requirements, number of artefacts, number of traces, precision of traces, size of artefacts, cost/effort of trace identification and maintenance, and value of traces (Heindl and Biffel, 2005).

The **value-based requirements tracing method** has five steps:

1. **Requirements definition:** The project manager or requirements engineer analyzes the software requirements specification and identifies atomic requirements. The requirements engineer then assigns a unique identifier to every requirement. The result is a list of requirements and their IDs (Heindl and Biffel, 2005).
2. **Requirements prioritization:** All stakeholders assess the requirements and estimate the value, risk, and effort of each requirement. The result of this step is an ordered list of requirements where the requirements are ranked on three priority levels (Heindl and Biffel, 2005).
3. **Requirements packaging:** This is an optional process step that allows a group of architects to identify clusters of requirements. These clusters are needed to develop and refine an architecture from a given set of requirements (Heindl and Biffel, 2005).

4. **Requirements linking:** The project team establishes traceability links between requirements and artifacts. Important requirements are traced in more detail than less important requirements. Therefore, we use 3 levels of tracing intensity. The result of this step is an overall traceability plan (Heindl and Biffel, 2005).
5. **Evaluation:** The project manager can use traces for certain purposes, e.g., to estimate the impact of change for certain requirements (Heindl and Biffel, 2005).

2.5.4 Agile development and traceability

Agile development is a software development style that offers a different approach from the traditional style by development practices that support frequently changing requirements, plans, and deliverables (Cockburn and Highsmith, 2001; Lee *et al.*, 2003). Agile development approach “*excels in exploratory problem domains - extreme, complex, high-change projects - and operates best in a people centred, collaborative, organizational culture*” (Cockburn and Highsmith, 2001).

Agile advocates for alternatives to the traditional way of documentation, contract negotiation and processes. Agile manifesto is read as follows:

“We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value [the following]:

1. *individuals and interactions over processes and tools*
2. *working software over comprehensive documentation*
3. *customer collaboration over contract negotiation*
4. *responding to change over following a plan*

That is, while there is value in the items on the right, we value the items on the left more”.

(Glazer *et al.*, 2008)

Requirements traceability practices can be applied to agile projects successful if it starts as early as possible in the development life-cycle (Lee *et al.*, 2003). Lee *et al.*, 2003, proposed to use Echo which is “*a tool-based approach designed to leverage the activities carried out by agile development teams to deliver the benefits of requirements engineering and traceability practices*” (Lee *et al.*, 2003). Echo first provides an effective means to record and share information gathered during requirements analysis, and secondly transparently delivers the benefits of requirements engineering and design rationale traceability, while maintaining true to the principles of agile methods (Lee *et al.*, 2003). Echo adheres to the principles and practices espoused by agile development methods, hence its success in capturing traceability (Lee *et al.*, 2003).

2.6 PROPOSED MODEL OF REQUIREMENTS TRACEABILITY

Figure 2, below shows the proposed model of requirements traceability derived from the literature we have reviewed in this chapter.

The model depicts traces that should exist between requirements, product features and other artefacts. The requirements engineering model used should be able to help with documenting user requirements, handling change requests, allowing for traceability links and artefacts to be captured and facilitating communication between stakeholders. The traceability model repository is a data container for the traces, links, artefacts and data types that are to be captured during the project life-cycle. Outputs and data from the requirements engineering model should define the structures and contents of the repository. In between requirements engineering or project activities and the repository lies a traceability model which acts as a bridge connecting the two.

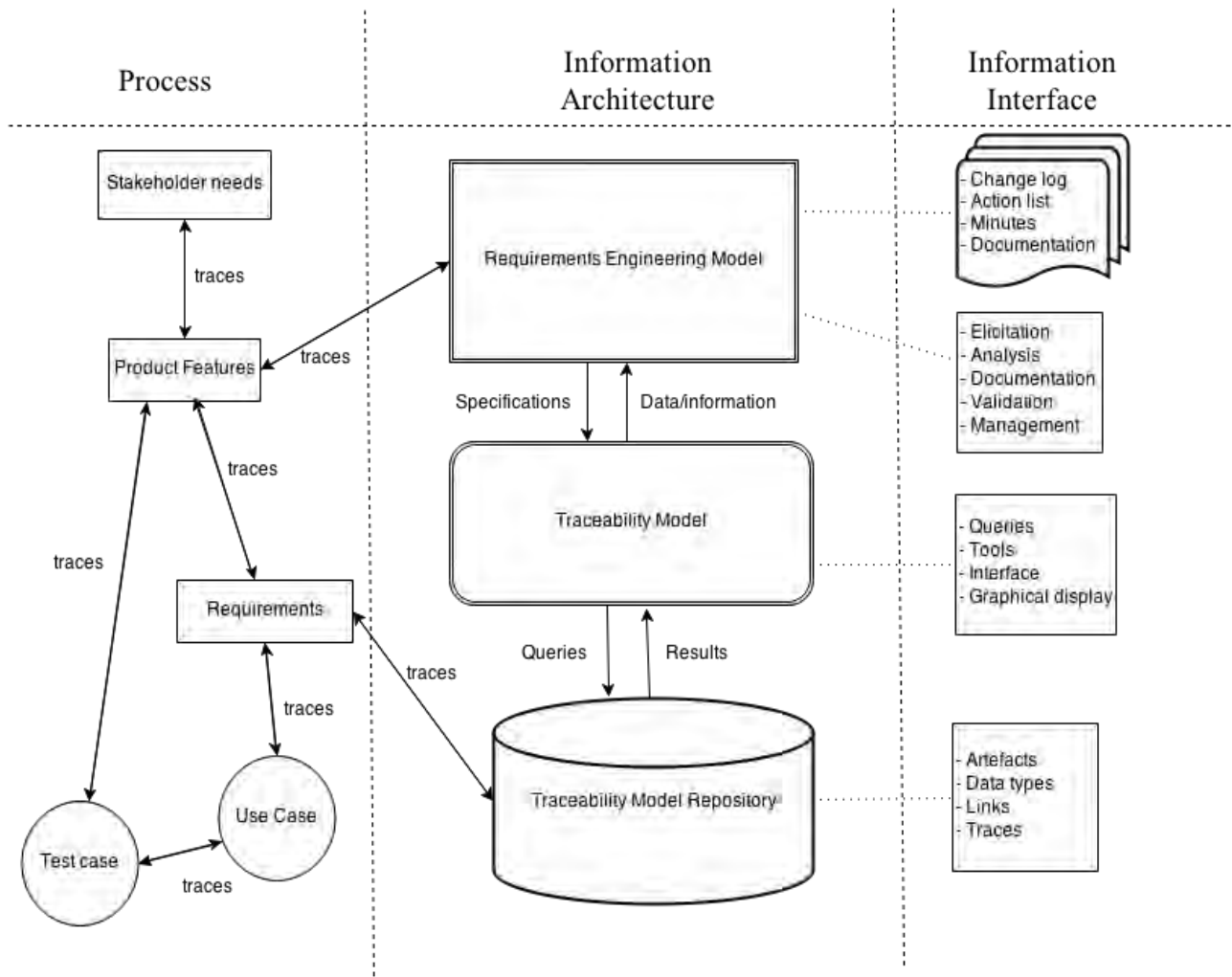


Figure 2: Proposed model of requirements traceability.

The model shows the kind of information and documents that can be produced and communicated to stakeholders. It acts as a view that allows certain piece of information to be retrieved. The model shows that each product feature should be subjected to a test case which also has traceable links.

Figure 3, below shows the proposed interaction of processes within the proposed model of requirements traceability. The model shows stages or phases in the project life-cycle. Stakeholder needs are mapped into requirements, and the process is iterative as more requests and changes come. Requirement and system engineering helps in determining functional, non-functional requirements and so forth resulting in requirements and system artefacts. All these processes from one stage to another are linked and the artefacts they produce are traceable.

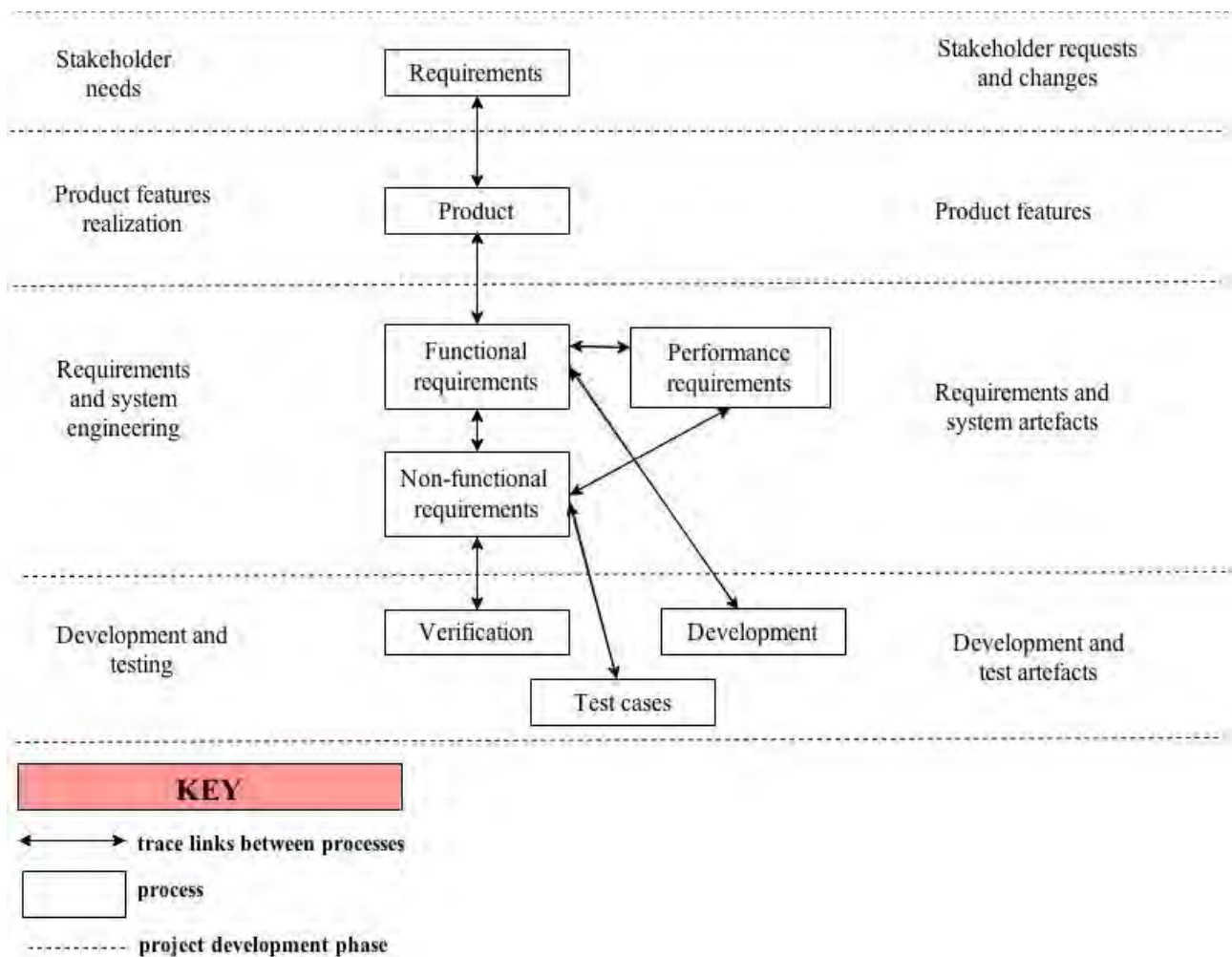


Figure 3: Proposed interaction of model processes.

2.7 CONCLUSIONS

The literature reviewed in this chapter has shown how requirements traceability is essential in requirements management practices. Software projects consists of interdependencies between various artefacts like requirements, design, source code and test cases (Heindl and Biffel, 2005). Traceability practices allows project stakeholders to manage these interdependencies in a transparent manner. It provides useful support for many software engineering activities such as requirements elicitation, requirements validation, change impact analysis and regression testing (Cleland-Huang *et al.*, 2004; Heindl and Biffel, 2005). It is useful in ensuring improved product quality and effective responses to changes. Many organizations have mandated, embraced and adopted requirements traceability practices as a measure of system quality (Ramesh, 1998).

Despite the recognition that requirements traceability practice has received over the past two decades, there are wide variations in the quality and usefulness of the practice (Ramesh, 1998). A major issue is that there is no agreement on a model of traceability that can be adopted across all projects and industries. Existing models, tools and techniques are too complex and the effort required to maintain traces is often too costly (Heindl and Biffel, 2005).

CHAPTER THREE: RESEARCH METHODOLOGY

3.1 INTRODUCTION

Research methods are classified in various ways and one of the most common distinctions is between qualitative and quantitative research methods (Myers, 1997). They are concerned with the research question and purpose, data collection, data interpretation, the write up, integrity measures, and inferences drawn (Bliss and Rocco, 2013). Depending on the type of data that need to be collected, the two methods can be used separately or both in a research.

A clear distinction between the two research methods is given as follows: **Quantitative research**, “Involves the numerical representation and manipulation of observations for the purpose of describing and explaining the phenomenon that those observations reflect.” (Abawi, 2008:10). **Qualitative research**, “Involves the examination and interpretation of observations for the purpose of discovering underlying meanings and patterns of relationship.” (Abawi, 2008:10).

This research study will focus on understanding requirements traceability issues in software development organizations using a qualitative research approach. Multiple case studies will be used as the research strategy of inquiry and data will be collected through semi-structured interviews. Semi-structured interviews will be conducted with selected software development and IT organizations in Cape Town. Multiple case studies are more appropriate for testing the theoretical model presented in this research, for establishing generalizability of findings, and for developing richer and more nuanced interpretation of a phenomenon (Bhattacharjee, 2012).

This chapter will provide and explain the justification for using qualitative research approach, multiple case studies and semi-structured interviews in this research study. It will also explain and elaborate on the research design, data collection methods and how the data is analysed and presented.

3.2 WHY QUALITATIVE RESEARCH?

The goal of this research is to seek deeper understanding, from multiple perspectives about how software development organizations trace requirements in their projects. In alignment with this goal are the assumptions enshrined within qualitative research. The assumptions are as follows:

1. Multiple realities exist in any given situation.
2. The research is context-bound.

3. Research is based on inductive forms of logic, categories of interest emerge mainly from informants.
4. The goal is to uncover and discover patterns of theories that help explain a phenomenon of interest.
5. Researcher interacts with those he/she studies and actively works to minimize the distance between the researcher and those being researched.

(Abawi, 2008).

In order for one to gain a better understanding of a phenomenon being investigated, it is important to understand its environment through interacting and empathising with its actors, as well as interpreting the actions and perceptions of these actors (Brockington and Sullivan, 2003). Software development is complex and difficult due to technical issues, the awkward intersection of machine and human capabilities, and the central role of human behaviour in software development activities (Seaman, 1999). It presents a number of unique management and organizational issues that are due to human behaviour, which is a complex phenomena that requires qualitative methods to study it (Seaman, 1999). This study is based on the premises of constructivism which concentrate less on verifying theories, but more on understanding how different people make sense of the world, and how they assign meaning to actions (Easterbrook *et al.*, 2008).

Qualitative research in this study will enable the researcher to collect richer, in-depth data, which will provide insights into requirement traceability issues in software projects that quantitative research approaches might not cover or provide enough information about (Abawi, 2008). The strengths of qualitative research methods lie in their usefulness for understanding the meaning and context of the phenomena studied, and the particular events and processes that make up these phenomena over time, in real-life, natural settings (Kaplan and Maxwell, 2005). These contextual issues in software development organisations include political, cultural and organizational concerns on what processes to follow in projects.

The justification for using qualitative research methods, as opposed to quantitative research in this study stems from the notion that qualitative research methods are designed to help researchers understand people and the social and cultural contexts within which they exist (Myers, 1997). Further justification arises because qualitative research methods are founded on an understanding of research as a systematic and reflective process for development of knowledge that can be contested and shared, and transferred beyond this study setting (Malterud, 2001).

The data obtained from qualitative research is much richer and informative than that obtained from quantitative methods. This makes qualitative research approaches a more suitable way of answering the questions posed in this study. Qualitative research methods involve the systematic collection, organisation, and interpretation of textual material derived from speech or observation (Malterud, 2001). It involves the use of qualitative data, such as interviews, documents, questionnaires and participant observation data, to explore the meaning of social phenomena as experienced by individuals themselves, in their natural context (Myers, 1997; Malterud, 2001). Qualitative methods force the researcher to dig deeper into the complexity of the problem rather than abstract it away (Seaman, 1999).

Software development environments differ from one organisation to another, and as such understanding these environments and their dynamics requires a qualitative approach of analysing things. Tools, models, techniques, procedures and technologies used in software development environments are subjective as they depend on organization's culture, strategy, preferences, opinions and nature of business. The use of qualitative research approach and multiple case studies as a means of inquiry in this research study will provide answers to the questions posed.

3.3 WHY MULTIPLE CASE STUDIES?

It is important for a researcher to decide on the most suitable research method and the sources of data for the research to be carried out. A research method is a strategy of inquiry which moves from the underlying philosophical assumptions to research design and data collection (Myers, 1997). The choice of research method influences the way in which the researcher collects data (Myers, 1997).

There are various research methods that can be used in qualitative research and these are:

1. *Action research,*
2. *Case study,*
3. *Ethnography and*
4. *Grounded theory.*

(Myers, 1997).

The research strategy used in this study is the use of multiple case studies. Case studies examine a phenomenon in its natural setting, employing multiple methods of data collection to gather information from one or a few entities (Myers and Avison, 2002). Understanding requirements traceability issues in multiple cases of software development organizations will provide a better understanding of similar instances and address the problems and issues identified (Sayre, 2001).

Studying multiple cases enables the researcher to become acquainted with the generality of requirements traceability in software development organizations and their conduct, procedures, tools, and models of traceability and to produce documents for use by the organizations under study and similar organizations (Sayre, 2001).

The reason why multiple case studies are used in this study instead of one case study is to get a more general understanding of requirements traceability in software development. Case studies are conducted to investigate a single entity or phenomenon within a specific time space and are suitable for software development and engineering environments because they avoid scale-up problems (Wohlin *et al.*, 2003). This study is an extension of the research that was done by Kanjanda (2012), who used a single case study when investigating the nature and extent of adoption and application of requirement management best practices in software development projects.

Case studies rely on multiple sources of information such as observations, interviews, collateral materials, corporate documents, and financial statements to provide an in-depth portrait of an organization or situation under study (Sayre, 2001). Multiple case studies are chosen in this study because they may lead to better understanding of major issues in requirements traceability. The following are benefits and strengths associated with case studies:

1. They can document multiple perspectives, explore contested viewpoints, and demonstrate the influence of key stakeholders and interactions between them in telling a story of the programme or policy in action. (Simons, 2009)
2. They provide descriptions of a situation faced by an organisation and bring the corporate environment to practitioners and students who cannot personally investigate issues. (Sayre, 2001)
3. Case studies enhance the application, testing, or generation of a theory. Researchers enter the case environment with a question in mind, which may or may not change during the course of preparing the case analysis (Sayre, 2001)
4. They are useful for exploring and understanding the process and dynamics of change. (Simons, 2009)
5. Case studies incorporate a variety of data collection and analysis activities (Sayre, 2001).
6. Case study using qualitative methods in particular enables the experience and complexity of programmes and policies to be studied in depth and interpreted in the precise socio-political contexts in which programmes and policies are enacted. (Simons, 2009)
7. Case studies are flexible, that is, neither time-dependent nor constrained by method. It can

be conducted in a few days, months or over several years and be written up in different forms and lengths appropriate to the time scale. (Simons, 2009)

8. They have the potential to engage participants in the research process. (Simons, 2009)

3.4 RESEARCH DESIGN

Research design is concerned with the strategy for collecting and analysing empirical data that will make it possible for the researcher to address the questions or research problem posed (Flick *et al.*, 2004). It constitutes almost all aspects of the research from collection and measurement of data to its analysis. The goals of this research study are only achievable if the research design is properly articulated, as such research designs are a means of achieving the goals of the research (Flick *et al.*, 2004). They link theoretical frameworks, questions, research, generalization and presentational goals with the methods used and resources available under the focus of goal-achievement (Flick *et al.*, 2004). Figure 4, below show the components of qualitative research design as given by Flick *et al.*, 2004.

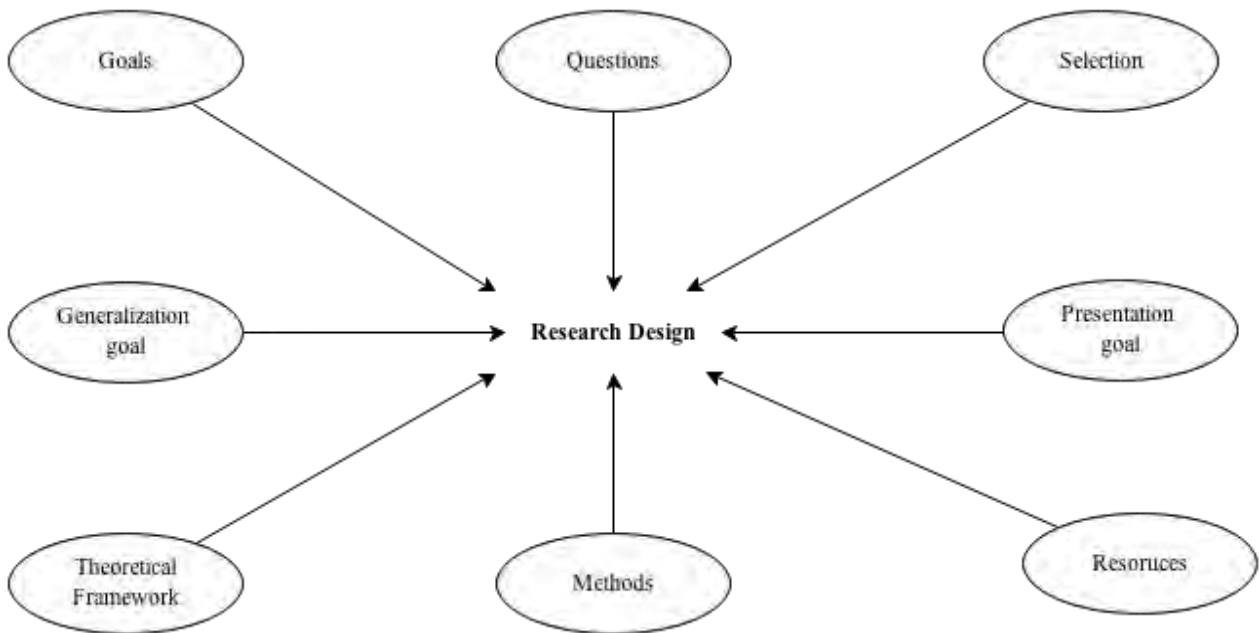


Figure 4: Components of qualitative research design.

Qualitative research design offers considerable flexibility in this study for the following two reasons:

1. Many aspects like processes, procedures and stakeholders of the software development projects change over time and as they change, the research study itself also may need to change (Kaplan and Maxwell, 2005).

2. Qualitative inquiry is inductive and often iterative in that the researcher may go through repeated cycles of data collection and analysis to generate hypotheses inductively from the data (Kaplan and Maxwell, 2005).

The selection of cases is a crucial step in case study research (Easterbrook *et al.*, 2008). To obtain a deeper understanding and greater validity (Easterbrook *et al.*, 2008), this research study will consist of 5 to 7 independent cases with software development companies in Cape Town. It is also important when doing research to choose an appropriate unit of analysis, to ensure the study focuses on the intended phenomena (Easterbrook *et al.*, 2008). In software development, the unit of analysis might be a company, a project, a team, an individual developer, a particular episode or event, a specific work product, processes and so on (Easterbrook *et al.*, 2008). This research study aims to interview project managers, software developers and/or senior management within the selected software development organizations. The unit of analysis in this study is the requirement management process.

3.5 DATA COLLECTION

Data collection refers to ways in which information can be obtained from the real world, recorded in a systematic way, quantified and/or explained (Cynthia *et al.*, 2004). It is typically directed toward discovering the **who**, **what**, and **where** of events or experiences, or their basic nature and shape (Sandelowski, 2000). The way data is collected in a study research contributes significantly to the quality, reliability, validity and credibility of the research (Myers and Avison, 2002; Farquhar, 2012).

Qualitative data sources include interviews, fieldwork, questionnaires, documents and texts, and the researcher's impressions and reactions (Myers, 1997). Data can also be collected from written data sources that include published and unpublished documents, company reports, memos, letters, reports, email messages, faxes, newspaper articles, social media posts and so forth (Myers, 1997).

This research study will use semi-structured interviews as data collection method. The semi-structured interviews will follow an interview guide, which consists of questions and topics that will be covered during the interview sessions. However, the interview will allow for flexibility and the ability to adapt to the context (Farquhar, 2012).

The structure of the interview, the questions and topics will be left to the interviewer's discretion.

The interview will consist of open-ended and specific questions, designed to elicit not only the information foreseen, but also unexpected types of information (Seaman, 1999) and these questions will be prepared ahead of time. Preparing the interview ahead provides well-prepared interview protocols that yield the best information and results that are depended on insightful interpretation (Sayre, 2001).

As mentioned early, the informants or interviewees in this research study will be project managers, software developers and senior managers in software development companies. The selection of these interviewees bodes well with the goals of this research, as they are the people involved in handling and transforming user requirements in software projects. These are the key personnel that may provide in-depth understanding of perspectives and may uncover organizational conflicts and issues pertaining requirements traceability in software projects (Sayre, 2001).

Interview data will be recorded in field notes and audio recordings to ensure accuracy of data when it comes to analysis (Seaman, 1999). Before the interview commends the interviewee has to consent to the use of audio recording.

3.6 DATA ANALYSIS AND PRESENTATION

Once the interviews have been conducted, recorded and transcribed, they have to be analysed and interpreted in order to write up the research findings and present them accordingly (Corbetta, 2003). The purpose of data analysis is to develop an understanding or interpretation that provide answers to the *who? what?*, and *where?* questions. (Kaplan and Maxwell, 2005). Data analysis is an iterative process that continues until an adequately coherent interpretation is reached (Kaplan and Maxwell, 2005). Data analysis is depended on judgement and interpretation of the researcher (Kaplan and Maxwell, 2005).

There are various techniques that a researcher may use separately or in combination to identify themes; develop categories; and explore similarities and differences in the collected data, and relationships among them (Kaplan and Maxwell, 2005).The following are the techniques that are used in qualitative data analysis:

1. *Coding,*
2. *Analytical memos,*
3. *Displays, and*
4. *Contextual and narrative analysis.*

(Kaplan and Maxwell, 2005).

3.7 LIMITATIONS

The limitations in this research study are:

- **Resource constraints:** Conducting interviews is a resource-demanding data collection method; activities such as planning, conducting and analysing are time-consuming by nature (Hove and Anda, 2005).
- **Case selection:** It will be difficult to select the right cases that will provide enough information to cover all aspects of requirements traceability.

CHAPTER FOUR: RESEARCH FINDINGS AND DATA ANALYSIS

4.1 INTRODUCTION

This chapter will discuss and analyse the findings of this research study. Data analysis will seek to identify similarities and differences that exists in software development and IT organizations when it comes to tracing requirements. This analysis will also use the proposed model of traceability presented in the previous chapter and theory from literature as a basis of comparison and generalizing the empirical data gathered in this research. Data analysis helps the researcher to get a total impression of the phenomenon under study, identifying meaningful units within the data and deducing meaningful conclusions (Malterud, 2001). The results of this analysis will lay the foundation for what this research study will eventually put across as conclusions and recommendations.

The strategy and techniques used in this research study for this data analysis will be briefly explained. A brief background of each organization that participated in this research will be given. For confidentiality purposes, the organisations will be identified by letters of the alphabet from A to E. The rest of this chapter will present the analysis of empirical data of this research study. Data will be analysed in relation to the topics set out in the interview process and these are: Requirements management, requirements traceability, requirements to product mapping and requirement change management.

4.2 DATA ANALYSIS TECHNIQUES

Data analysis consists of examining, categorizing, tabulating testing qualitative evidence to address the initial propositions of a study (Perry *et al*, 2004). As described earlier, data analysis is an iterative process that continues until an adequately coherent interpretation is reached and is depended on judgement and interpretation of the researcher (Kaplan and Maxwell, 2005). The main goal of data analysis is “*the search for coherence and order*”, Kaplan and Maxwell (2005). Analysing qualitative data is essentially about detection, and the ability to define, categorizing, theorize, explain, explore and map the data are fundamental to the researcher (Bryman and Burgess, 2002).

Pope *et al*, (2000) identified five stages that can be used in analysing qualitative data:

- **Familiarisation** - studying interview notes and listening to the recordings to capture popular themes and important notes (Pope *et al*, 2000).

- **Identifying a thematic framework** - identifying all the key issues, concepts, and themes by which the data can be examined and referenced (Pope *et al*, 2000).
- **Indexing** - applying the thematic framework or index systematically to all the data in textual form by annotating the transcripts with numerical codes from the index (Pope *et al*, 2000).
- **Charting** - rearranging the data according to the appropriate part of the thematic framework to which they relate, and forming charts (Pope *et al*, 2000).
- **Mapping and interpretation** - using the charts to define concepts, map the range and nature of phenomena, create typologies and find associations between themes with a view to providing explanations for the findings (Pope *et al*, 2000).

There are five specific analytical techniques intended to deal with problems of developing internal and external validity in doing case studies and these are:

- Pattern matching,
- Explanation building,
- Time-series analysis
- Logic models
- Cross-case synthesis

(Perry *et al*, 2004)

In order to gain an understanding of the proposed model of traceability used in this research, pattern matching and cross-case synthesis are the two techniques employed to seek evidence that test the model. Through pattern matching, empirically observed patterns within each case studied can be compared with patterns in other cases or patterns established in previous studies and in different contexts (Gibbert *et al*, 2008). Pattern matching compares an empirically based pattern with a predicted one and if the patterns coincide, the results can strengthen the internal validity of the case study (Perry *et al*, 2004).

Cross-case synthesis is used where at least two cases are studied (Perry *et al*, 2004). Case study research can be used to achieve various research aims: to provide descriptions of phenomena, develop theory, and test theory (Darke *et al*, 1998; Gibbert *et al*, 2008). A cross-case synthesis involving many case studies may provide a good basis for analytical generalization - from empirical observations to theory (Gibbert *et al*, 2008). Cross-case synthesis enhances generalizability and helps to deepen understanding and explanation (Miles and Huberman, 1994).

Explanation building analyses case study data by building an explanation about the case and it is a special type of pattern matching but the procedure is more difficult to follow (Perry *et al*, 2004). Time-series analysis match observed trend with pre-defined trend and comes in various forms and can be complex and problematic when collecting data (Perry *et al*, 2004). Logic models are an analytic technique that consists of matching empirically observed events to theoretically predicated events (Perry *et al*, 2004). Explanation building, time-series analysis and logic models are resource intensive and time consuming techniques and for this reason they will not be considered in this research study.

4.3 PARTICIPATING ORGANISATIONS

The organizations that participated in this research were first invited through email and telephone conversations. Subsequent communication was through emails in which, the confidentiality and consent letter and the cover letter (see Appendix A and Appendix B) were send and appointments for interviews were made. Before the interview started the researcher explained to all participants what the research study was about and what it seek to achieve. At that moment the confidentiality and consent letter was signed by the participant and the researcher (see Appendix C for signed letters). The participants were also presented with the proposed model of requirements traceability presented in the previous chapter. The organizations and their respective participants will not be addressed by their true names for confidentiality purposes.

4.3.1 Organisation A

The organization is a Microsoft Gold Certified Partner founded in 1997 and specialise in creating custom software solutions, integrating existing client applications and solutions and customisation of packaged applications and solutions to their client needs. The organization serve big retail banking institutions such as Standard Bank, ABSA and Barclays, government institution like SARS and other companies like British Petroleum. The company has branches in Cape Town, Johannesburg and London and it has more than 200 employees. The participant was the Product Manager, who once served as the Project Manager for 4 years before his current role.

In their own words from the interview

“Often requirements are fairly big in volume and client organizations do not understand their own requirements in most cases, so the organization decided to be involved right upfront when the requirements are being gathered and throughout the project life-cycle.”

4.3.2 Organisation B

The organization was formed in 2007 by two former University of Cape Town students and is based in Gardens, Cape Town. It offers end-to-end software and project management services including analysis, design, development, and maintenance and server administration. It creates and maintains bespoke online business solutions, gaming platforms and web based application for insurance companies like Direct Axis, MiWay and Hippo. Other companies it has done work for include recruitment company PNET and online gaming company based in the United Kingdom. At the time of the research interview the company had 25 employees comprising of project manager, developers, designers and search engine optimizers. The participant was the Project Manager who has been working for the companies for 2 and half years.

In their own words from the interview

“The nature of project management is forever changing, and the tools and methods used by the organization are useful and serve the purpose.”

4.3.3 Organisation C

The organisation is a Microsoft Silver Certified Partner founded in Johannesburg in 1987. It has been a Microsoft Silver Certified Partner since 2007 and is proud to have the silver awards across the competencies of SQL Server, C# and ASP.Net. The company head office is in Cape Town, with other offices in Gauteng. The company has around 45 IT professionals, from Scrum Masters and Business Analysts to Web, Mobile and Desktop Developers. The company’s core values are those of honesty, integrity and respect with a firm commitment to innovation, openness and delivery. The company has provided software solutions to various industries such as retail, entertainment, logistics and insurance. The participant was the head of software development team in Cape Town and has worked for the company for 4 years.

In their own words from the interview

“SCRUM is not very strict on requirements or documents traceability and focuses on a collaborative approach with various stakeholders as well as allowing requirements to not be too prescriptive upfront but rather allowing them to develop and change through the duration of a project.”

4.3.4 Organisation D

The organization was formed in 2004 in Macedonia. It established its Cape Town office in 2009. Its

software development philosophy is informed by ideas enshrined in agile development manifesto. The organization has 17 employees including developers, designers, system architects and interns. It has provided online and mobile solutions to various companies including Non-Governmental Organizations, e-learning services and Internet Service Providers like Hertzner. The participant was the senior developer who has worked for the company since 2009 and has more than 6 years' experience in software development.

In their own words from the interview

“The human element of managing expectations is far more important than managing processes. Agile development prescribes people and communication over tools and processes.”

4.3.5 Organisation E

The organization was formed more than nine years ago in Cape Town. It has been delivering software consulting and software development services to leading businesses in retail and commercial industry. The organization has 25 employees including developers, designers and systems analysts. The organization develops web applications and mobile applications that help their clients to successfully communicate with and interact with their consumers. They have done applications for the Google and Apple app stores. The participant was the application development manager who has been with the company for 6 years.

In their own words from the interview

“Requirements are always difficult to manage because clients keep on changing their minds. The organization participate in every requirement management process so that it keeps abreast of all the changes to requirements”

4.4 ANALYSIS OF RESEARCH DATA

4.4.1 Overview of findings

In data analysis it is important to first identify themes and patterns that helps the researcher to develop categories, and explore similarities and differences in the data, and relationships among them (Kaplan and Maxwell, 2005). The analysis will start by giving an overview of how the participating organizations handle and manage requirements. Table 3, below shows an overview of how each organization performs its requirements related activities and processes in relation to requirements management, requirements traceability, requirements to product mapping and

requirements change management.

Organization	Requirement management processes	Requirement traceability	Requirements to product mapping	Requirements change management
A	Formal processes defined, not strictly followed. Requirements documents exists.	Requirements traceability exists. Not always planned. It is a deliverable. No automation tools.	Difficult to achieve but test cases exists. Acceptance tests. Code comments, and requirement scenarios.	Formal processes defined. Risk assessment. Version control mechanism. Authorizations done in mutual agreement
B	No formal processes defined. Requirements documents exists.	Requirements traceability exists. It is not planned. No automation tools.	Test cases exists. Code comments and functions easy to understand	No formal processes defined. Version control mechanism. Clients authorize changes.
C	Formal processes defined but not strictly followed. Requirements documents exists.	Requirements traceability exists but not prioritised. No automation tools.	Test cases exists.	No formal processes defined. Version control mechanism. Changes agreed and authorized in SCRUM
D	Formal processes defined but not strictly followed. Requirements documents exists.	Requirements traceability exists but not prioritised. No automation tools.	Test cases exists. Behavioural drive IT, test-driven development.	No formal processes defined. Version control mechanism. Senior team member and clients can authorize change.
E	Formal processes defined but not adhered to. Requirements documents exists.	Requirements traceability exists. It is not planned. No automation tools.	Test cases exists. Development environment enforces tests	Formal processes exists. Version control mechanism. Changes authorized mutually in meetings.

Table 3: Summary of responses to major aspects in requirements management.

The participating organizations were asked several questions pertaining to requirements (see Appendix D and also see Appendix E for the interview responses). The main themes that emerged from the interviews include; the presence of formal processes for requirements management, understanding of requirements traceability as a core process of requirements management, presents of requirements documentation, existence of test cases and test environments and the presents of formal processes to handle changes to requirements. Table 4, below gives an overview of the existence of requirements management processes and associated documents in the participating organizations' projects environment.

Description	A	B	C	D	E
The organisation has formal processes to support requirements management	Yes	No	Yes	Yes	Yes
Organization is involved in all the requirements management processes	Yes	Yes	Yes	Yes	Yes
Organization understand requirements traceability and the role it plays in projects	Yes	No	No	Yes	Yes
The organisation keep formal requirements documents such as requirements matrix	Yes	No	No	No	Yes
The organization has a database (excel, online resources) where requirements are stored and can be traced.	Yes	Yes	Yes	Yes	Yes
The organisation has test cases in its development environment to test all products requirements and functions	Yes	Yes	Yes	Yes	Yes
The organization has formal processes to handle changes to requirements	Yes	No	Yes	No	Yes
The organization's overall requirement management processes is effective and efficient for them to produce quality projects	Yes	Yes	Yes	Yes	Yes

Table 4: Overview of existence of requirements management processes and documents.

As mentioned in Chapter One, the purpose of this study is to investigate the extent to which requirements traceability affect the project quality, success and eventual organization's performance. Table 5 below, gives an overview of organisations value of requirements management processes and tools in relation to project quality and organization performance.

		Formal processes	Requirements documentation	Traceability tools and models	Requirements change management	Test cases and version control
A	VQ	✓	✓	✓	✓	✓
	VP	✓	✗	✓	✓	✓
B	VQ	✗	✓	✓	✓	✓
	VP	✗	✗	✗	✓	✗
C	VQ	✓	✓	✓	✓	✓
	VP	✓	✗	✗	✗	✗
D	VQ	✓	✓	✓	✓	✓
	VP	✓	✗	✗	✓	✗
E	VQ	✓	✓	✓	✓	✓
	VP	✓	✓	✗	✓	✓

Key

VQ – Value on project quality and success

VP – Value on overall organisation performance

Table 5: Organisations value of requirements management processes and tools in relation to project quality and organization performance.

Ramesh (1998) identified two types of requirements traceability users; low-end and high-end users. This research will extend from the concept used by Ramesh but will redefine the traceability users as follows:

- **Inactive user** - views requirement traceability as not very important in their projects but they have to some extent resources to maintain and manage their requirements.
- **Dormant user** - views requirement traceability as important in their projects but the mandate lies with the project sponsor or clients. They have resources to maintain and manage requirements.
- **Active user** – views requirements traceability as a critical aspect of the requirements engineering process and is a deliverable. They have resources to maintain and manage the requirements.

Table 6, below shows how the organisations which participated in this research study are classified

in terms of the traceability users define above.

Traceability user	Organisation	Description
Inactive user	Organization B	No formal processes of managing and maintaining requirements. Requirements documentation exists but requirements tracing is not a priority. There is no office of project manager, and traditional project management principles are not well known.
Dormant user	Organization C, Organisation D	Formal processes of managing and maintaining requirements are defined. Requirements documentation exists but requirement tracing is a mandate of the client or sponsor. Modern methods of software development like Agile development are preferred in project environments to the traditional waterfall model.
Active user	Organization A, Organization E	Formal processes of managing and maintaining requirements are defined. Requirements documentation exists and requirements tracing is planned and it is a deliverable. All stakeholders value requirements tracing. Modern methods of development can be used together with traditional methods.

Table 6: Organisation classification in terms of traceability users.

Although Organisation B uses Google docs, Trello and excel sheets to handle and manage their requirements, they do not have formal processes established. These documents for them are not deliverables, hence they are classified as inactive traceability users. Organisations C and D, are dormant users because they follow religiously the agile development philosophy, which does not value processes and documentation. They use agile development applications like JIRA and VirtualTracker to maintain and manage their requirements. Although Organisation A appreciates and use agile development methods in some of their projects, they always plan for requirements traceability. They believe in mixing traditional and modern methods of software development. They have formal requirements management processes and in most of their projects requirements traceability is a deliverable. They are an active traceability user. Organization E uses Basecamp and the traditional requirements traceability matrix to manage and maintain their requirements. The fact that they are sometimes required to produce requirements traceability as a deliverable by clients make them an active traceability user.

4.4.2 Requirements and requirements management processes

4.4.2.1 General overview in theory

Theory suggests that requirement engineering should be carried out in projects and documents created and maintained should be traceable with links well established (Wieringa, 1995; Lee *et al*, 2003). Requirements engineering processes of elicitation, analysis, documentation, validation and management are essential in the success of projects. Data and information produced at each process level should be well documented and all artefacts should be maintained. The proposed model of requirements traceability presented in this research indicates that requirements documents should be produced and traces should be stored in a database.

4.4.2.2 Empirical study - common to most organisations

- Organizations A, C, D and E have formal requirements management processes, but they do not strictly follow them. Organization A maintains requirements documents in the form of screen mark-ups, design artefacts, wire-frames, issue documents, traceability matrix and minutes of meetings. Although Organization B does not follow a formal requirements management process, they maintain requirements documents in the form of Google Docs and excel spreadsheets. Organizations C and D use agile development tools like user stories, pictures and white boards to manage and maintain their requirements. Organization E just like organization A, uses a mixture of both worlds, Basecamp is a more modern tool used as a collaboration and communication tool. It stores all the requirements and documents on the web. They also use requirements traceability matrix for some projects.
- All the organizations are totally involved in all the requirements management processes, of elicitation, analysis, documentation, validation and management. They all agreed that participating in these processes is of paramount importance and proper management of requirements is critical to the success of the project. Organization D pointed out that most of the architectural decisions they make are influenced by the actions and words of the stakeholders during a SCRUM sessions, hence the need to be involved upfront.
- There is a general agreement from all the organizations that clients do not quite understand requirements so more often they change their minds. Organizations E pointed out that clients always change their minds that is why they need to be involved in all the requirements management processes.

4.4.2.3 Empirical study – Major differences amongst the organisations

- Although all the organizations maintain all sorts of documents, there is no one similar way

of maintaining and managing the requirements. Organization A uses requirements matrix, wire-frames, screen mark-ups and so on, while Organization B uses Google Docs and excel spreadsheets. Organization E uses spreadsheets, requirements matrix and Basecamp. Organization C and D use user stories and project backlogs to maintain and manage their requirements.

4.4.3 Requirement traceability

4.4.3.1 General overview in theory

Requirements traceability is concerned with relating requirements specifications with other artefacts created in the development life-cycle of a software system (Spanoudakis *et al.*, 2004). Mader *et al.*, (2013) identified six practices for strategic traceability in safety-critical projects which are also applicable to software projects as plan traceability, offer traceability tool support, create traceability incrementally, model traceability queries, visualise trace slices and evaluate traces continually. The following section will report the findings of this research in comparison with what theory has suggested in terms of traceability practices in projects. The model of requirements traceability presented in this research shows that requirements should be traceable, with documents, artefacts and links well established.

4.4.3.2 Empirical study - common to most organisations

- There is some understanding from all the organizations on what requirements tracing is all about. There is a general sense and understanding from all the organization that requirements should be tracable back to their original source at any stage of the project life-cycle. Organization A regards requirements traceability as a means of delivering what the client has asked for and Organization D regarded it as the management of a requirement from project inception to the final product. In theory and literature, requirements traceability is bi-directional, from requirement to product and from product to requirement.
- Not all the organizations have formal procedures or processes of planning requirements traceability. Organization B pointed out that, requirements traceability is not really planned but it is vitally important to start thinking about how to manage and maintain the requirements right at the beginning of a project. Organization A only does planning for requirements traceability if it is a deliverable specified by the client.
- There is evidence from the research that all organizations keep some form of documents and design artefacts they use in tracing user requirements. Although each organization has its own way of storing and managing requirements, the use of spreadsheets is common to all the organizations. Organization A pointed out that spreadsheets are easy to work with

especially when a project and its requirements are small, however it's problematic when the requirements are more numerous.

4.4.3.3 Empirical study – Major differences amongst the organisations

- Organizations A, C, D and E have formal requirements management processes, but they do not strictly follow them. Organization A maintains requirements documents in the form of screen mark-ups, design artefacts, wire-frames, issue documents, traceability matrix and minutes of meetings. Organizations A and E use a traceability matrix selectively on projects depending on project and client needs. Organization B, C, D and E apart from the usual spreadsheets, they also use web based applications to store and maintain requirements. Web based application like Basecamp, Trello and VirtualTracker provides the necessary environment for organizations to manage their requirements. They make it easier for all stakeholders to collaborate and communicate.

4.4.4 Requirement to product mapping

4.4.4.1 General overview in theory

Requirement traceability is a quality factor and a non-functional requirements that a software system should possess (Ramesh and Jarke, 2001). Requirements verification and validation are essential processes in achieving software quality objectives. One way organizations can check for quality is by conducting tests in their project environments. The discussion below gives an analysis of the findings of this research in terms of testing and ensuring the requirements are mapped correctly to the product.

4.4.4.2 Empirical study - common to most organisations

- All the organizations run unit, functional and acceptance tests in their project environments. Tests are very important in projects because they indicate the extent to which a product being designed or developed meets the requirements as specified by the clients. Tests are a means to ensure product quality. Organization A pointed out that JIRA provides Test Case Management system which is flexible to use. Organization D carries out test driven development, which allows to conduct tests on a product as each feature is added. Each feature of a product has a test case defined for it.
- All the organizations conceded that it is difficult to indicate a specific requirement in the software code. They all agreed that comments are added to software code just to get an idea of what is being developed. Organization A highlighted that code commenting is manageable if requirements are few and the system being developed is not large. Function

and class definitions just give an idea of what the code does.

4.4.4.3 Empirical study – Major differences amongst the organisations

- Each organization has its own methods and tools of testing product requirements. Organization A uses JIRA. In Organization B test are done as the code is being implemented, they have a development server. All the product features are tested on the development server by developers and the clients. Organization D carry test driven development.

4.4.5 Requirement change management

4.4.5.1 General overview in theory

Requirements change refers to the emergence of new requirements or the modification or removal of existing requirements (Lam and Shankararaman, 1999). In this research the idea was to find out if organizations have formal processes to handle changes to requirements and if the changes have any effects to the traceability of these requirements. The analysis below reports on the findings of this research in this regard.

4.4.5.2 Empirical study - common to most organisations

- Most organizations have formal processes of managing changes to requirements. In Organization A the formality of the process is determined after a risk assessment of the change and depending on the severity of the change. In organization B and E changes are agreed upon in meetings that are conducted to review the projects. Organization C and D changes are done in SCRUM sessions.
- Changes to requirements are done in all organizations with the full consent of the client. Clients have a final say and they authorize these changes to requirements.
- All the organizations have version control mechanisms for their code and documents.

4.4.5.3 Empirical study – Major differences amongst the organisations

- Organization A and E use the traditional processes of requirement change management whereby a change request is raised and submitted for a review. They perform risk assessment before a final decision is made. Organization B, C and D use meetings and SCRUM sessions to handle changes to requirements.

4.4.6 Overall analysis of findings

All the participating organizations concluded that their requirements management processes are

efficient and effective. Organization A and E are actively involved in the requirements engineering processes and they maintain all sorts of documents to help manage and trace project requirements. The organizations are satisfied with the processes and methods they use to manage requirements although there is a general feeling that they are not adequate.

Organization B pointed out that the nature of project management is forever changing and the tools and methods they use to handle and manage requirements are useful and serve the purpose. They also indicated that the processes are effective and are helping to increase productivity and shorten the time a project is delivered to the client.

Organization C and D software development environments are deeply rooted in the agile development philosophy. The methodology value people management and communication over processes and documentation. However they maintain project documentation for contractual and accountability purposes. Organization D pointed out that keeping requirements traces helps in maintaining accountability.

Table 7, below give an overall analysis of the research findings in comparison with the processes, information architecture and information interface defined and illustrated in the proposed model of traceability presented in this research study. The model shows an emphasis on processes like the requirements engineering processes, requirement change management and running of test cases. It also presented an information architecture where requirements and requirements traces are stored. The information interface of the model shows all sort of documents and artefacts that produce effective and efficient traceability in projects.

Organization	A	B	C	D	E
Documents and artefacts	Requirements traceability matrix, JIRA reports, wire-frames, screen mark-ups and spreadsheets	Google Docs, Trello reports, spreadsheets	User stories, use cases, product backlogs and spreadsheets	User stories, use cases, product backlogs, Pivotal tracker charts and burn downs and spreadsheets	Requirements traceability matrix, use cases Basecamp reports and spreadsheets
Requirements engineering processes	Total involvement	Total involvement	Total involvement	Total involvement	Total involvement
Requirement management processes	Defined, but not strictly followed	No formal processes defined	Defined, but not strictly followed	Defined, but not strictly followed	Defined, but not strictly followed
Understanding of requirements traceability	Deep understanding	Not well understood but drive and passion exists	Little understanding and appreciation	Little understanding and appreciation	Deep understanding
Planning of requirements traceability	Planning done when necessary	No planning but thinking upfront	No planning	No planning	Planning done when necessary
Trace databases	JIRA provides facility for storing traces. Spreadsheets	Spreadsheets	Agile development tools. Spreadsheets	Agile development tools. Spreadsheets	Basecamp. Spreadsheets
How requirements traceability is achieved	Documents and design artefacts	Documents and design artefacts	Design artefacts, less documentation	Design artefacts, less documentation	Documents and design artefacts
Accessibility of information on requirements	Easily accessible, JIRA has easy interface for queries	Easily accessible, Google docs are on cloud computing	Easily accessible, most agile tools are web bases and cloud computing	Easily accessible, most agile tools are web bases and cloud computing.	Easily accessible, Basecamp is web based.

Value of requirements traceability to the organization	Valuable when dealing with third-party and is a deliverable.	Valuable in tracing root sources of problems. Boost productivity.	Very valuable to cover for unseen problems	Very valuable, helps in tracking performance.	Very valuable, it helps keep track of things in project.
Value of requirements traceability to their clients	Very valuable, clients use it to measure what they have asked for.	Very valuable because they keep changing their minds	Not very important to them they just want to see things working	Valuable because it builds trust and set obligations for doing something.	Very valuable to clients, they need it as a deliverable.
Formal processes to manage change to requirements	Formal processes defined	No defined processes, meeting resolves issues	No defined process. Client has the power.	No defined processes, anyone can authorize changes	Formal processes. Mutual agreement in meetings.

Table 7: Overall comparison of requirements traceability practices.

4.5 CONCLUSIONS

The analysis of empirical data done in this chapter, shows that there is a general understanding of what requirements traceability is all about and how valuable it is in project environments. Although most organizations are not willing to invest and spend money on it, they see its value. Organizations have various ways of handling and managing requirements, some are formalized processes and some are very informal but the overall goal is to produce quality products that meets the expectations of their clients.

The next chapter will give the conclusions of this research study by giving recommendations and pointing out areas that may need to be studied in the future in this field of study.

CHAPTER FIVE: CONCLUSIONS AND RECOMMENDATIONS

5.1 INTRODUCTION

This chapter will draw conclusions to this research by explaining and outlining whether the findings analysed in the previous chapter have addressed and answered the research questions. It will also outline whether the objectives and aims of this research study were achieved and fulfilled. Furthermore, it will conclude by giving recommendations and suggestions for areas of further research.

5.2 PROBLEM STATEMENT AND RESEARCH QUESTIONS

As a refresher, the problem statement in this research study was defined in chapter one as:

In software development and information technology projects, requirements constantly change and become more complex, these changes are not being adequately traced throughout the project life cycle resulting in project delays, high project costs, poor quality products and in most cases premature project termination.

The findings of this research shows that although organizations understand requirements engineering processes they still lack appreciation of requirements traceability. The findings has shown that requirements are not being adequately traced in most projects. Most organizations only start to think of requirements traceability when it is a deliverable as per client request.

In chapter one the two research questions to be addressed were stated as:

1. Are software and information technology companies tracing project requirements in both forward and backward directions?
2. Are there recognized requirement traceability models and tools that have been adopted for the industry, how effective and efficient are they and how are they helping in improving quality and project deliverance?

The next section is going to outline whether these two questions were addressed and answered by the findings of this research.

5.2.1 Tracing requirements in both directions

In literature and what has already been discussed in chapter two, requirements traceability was

defined as the ability to describe and follow the life of a requirement, in both forward and backward direction (Gotel and Finkelstein, 1994; Wieringa, 1995; Domges and Pohl, 1998; Jarke, 1998). The findings of this research showed that most organizations value requirements engineering processes and they want to be part and parcel of all activities concerning requirements. Although there is great involvement in these activities, organizations are not tracing requirements adequately in both directions.

Requirements traceability should be able to show the degree to which a relationship can be established between two or more work products of the development process (Kirova *et al.*, 2008). The organizations that participated in this research maintain all sorts of requirements documents and artefacts for managing and handling requirements. This documentation is in the form of requirements matrix, wire-frames, screen mark-ups, user stories, use cases, Google documents and online web applications like JIRA, Basecamp and Trello. Most of these documents and web applications are informal and are not able to generate meaningful trace links and relationships between work products.

Although to some extent the organizations have requirements traceability and some of them maintain formal documents like requirements matrix, there is no evidence of both forward and backward traceability.

5.2.2 Traceability models and tools

Chapter two presented some modern commercial tools and environments for requirements traceability as identified by INCOSE, 2004. These tools and environments are used in various industries and can be applied to software development and IT projects.

The findings of this research showed that organizations in software development and information technology projects are not willing to use any automation models and tools of requirements tracing. Most of the organizations clearly pointed out that they are not interested in these automation models and tools. Most commercial products are expensive hence some organizations indicated that they are always looking for new ways of managing their projects since the nature of project management is always changing. These new ways being used in software development and IT projects are helping in managing projects requirements and tasks but are not suitable and adequate for tracing projects requirements.

5.3 FULFILMENT OF AIMS AND RESEARCH OBJECTIVES

To recap what was discussed in chapter one, the aim of this research study was to determine how well the idea of tracing requirements is used in the software and information technology industry and what role it plays in producing a quality and successful project. The idea of tracing requirements is well known and highly valued in software development and IT industry but it's not adequately implemented. The findings here have shown that there is no planning involved when it comes to requirements tracing. Organizations only start thinking about it if it's a deliverable requested by a client.

In the literature requirements traceability has been identified as a quality factor and characteristic a system should possess and include as a non-functional requirement (Ramesh and Jarke, 2001). All the organizations are satisfied with the methods and processes they use in their project environments and they believe clients trust the quality of their processes and the products they produce.

The next section will discuss the findings in the context of research objectives. In chapter one the research objectives of this research were stated as follows:

1. Determine how requirements are collected and documented in software projects environments.
2. Determine how requirements are being traced in software projects.
3. Identify factors that are affecting requirement traceability in software projects.
4. Identify important aspects and issues that should be captured and addressed by requirement traceability.
5. Identify traceability tools and models that are being used in the software and IT companies in Cape Town.

1. Determine how requirements are collected and documented in software projects environments.

All the organizations that participated in this research are involved in requirements management processes. Requirements are collected in meetings and SRCUM sessions that are done with clients on a regular basis. Other forms of collecting requirements includes workshops, initial documents by the clients, prototypes and user stories from clients. As shown in the empirical findings, organizations have various methods, processes and platforms to record and manage their requirements. Pivotal Tracker, JIRA Agile, Basecamp and Trello are some of the applications and

platforms where requirements are kept. These tools are used to manage the requirements management processes including communication and validation

If there is need for change in the requirements, certain processes are followed and all the stakeholders are informed. In most organizations there are processes to manage requirements changes. Changes to requirements are depended on the severity of the change but most decisions on requirements are done by mutual consent.

2. Determine how requirements are being traced in software projects.

The findings of this research has shown that organizations are not willing to implement methods and processes that help them in tracing requirements. The tools and methods used by organizations help them to manage the management processes rather than the traceability of requirements. However, some organizations pointed out the tools and applications they use like JIRA Agile help them to create links and pointers to requirements documents like traceability matrix and design artefacts. Other applications like Google documents, Trello and Basecamp provides archives and historical reports that help the organizations to trace the movement of requirements. A good software configuration management system should be able to establish links and traces whenever changes are done to the project through providing version control, configuration identification, and support for accounting activities related to requirements change management (Cleland-Huang *et al.*, 2003). Most organizations have version control mechanisms in their development environments.

3. Identify factors that are affecting requirement traceability in software projects.

As highlighted earlier INCOSE, 2004 identified tools that give full requirements traceability. The research findings here have shown that none of the organizations use the tools identified. This is one major issue affecting requirements traceability practice in software projects. Most organizations prefer custom tools and models to commercial tools and models. Commercial tools and models provide full requirements management and requirements traceability.

Another factor affecting traceability practices in software projects from the findings of this research is the advent of agile development methods. Although some literature argued that agile development and the traditional methods can be used together in projects (Lee *et al.*, 2003), most organizations in this research prefer using agile development methods alone. The organizations highlighted that it is not necessary for them to worry about the traceability of requirements because agile does not advocate for documents and requirement tracing.

Organizational policy and culture are other contributing factors affecting traceability. The adaptation of tools, processes and methods of managing and maintaining requirements are simply decisions made by the organization in-line with their policy and culture. The organizations highlighted that decisions on tools, processes and methods used are purely organizational and are in-line with their beliefs.

Finally, cost in terms of time and effort required to create and maintain traceability documents like requirement traceability matrix is another factor affecting traceability practice. Organizations highlighted that this cost become too high if the project requirements keep on changing and the project itself keeps growing.

4. Identify important aspects and issues that should be captured and addressed by requirement traceability.

Requirement traceability should be able to take initial requirements and trace them throughout the project life-cycle, from design and architectural models, source code, validation and test cases. Literature has highlighted that requirement traceability aids in system comprehension, impact analysis, system debugging, and communication between the development team and stakeholders (Asuncion *et al.*, 2007). Requirements traceability should be able to capture the original source of requirements and create links and relationships between requirements and product artefacts. One aspect that the researcher found is that requirements traceability is used by organizations to address issues of trust and accountability between them and their clients. Another aspect found in this research is that design artefacts and requirements documents are used by organizations as part of the contract document.

5. Identify traceability tools and models that are being used in the software and IT companies in Cape Town.

The findings of this research has shown that most organizations do not use any commercial traceability tools and models. Organizations are using web applications to capture and maintain their requirements but these applications do not provide full requirements tracing. As mentioned earlier web applications and tools like Pivotal Tracker, JIRA Agile, Basecamp and Trello help the organizations to manage their requirements but are not sufficient to provide full requirements traceability. Most organizations highlighted that they are not interested in any automated tools and models of requirements traceability.

5.4 RECOMMENDATIONS AND AREAS OF FURTHER RESEARCH

The conclusions given here are not sufficient to cover the whole area of requirements traceability in software development and IT projects. As a result, the following further areas of further research study are recommended:

1. The majority of organization that participated in this research work in agile development environments. The agile development manifesto does not value requirements documentation and requirements tracing. The researcher recommends that further research studies be conducted to ascertain how requirements traceability can be implemented in agile development environments.
2. In chapter four Organization B was categorised as an inactive user of traceability. The reasons mainly being that they do not have formal processes to handle and manage requirements and that they do not employ a qualified project manager or project practitioner. However, the organization use methods and processes that make them succeed in producing quality products. The researcher recommends further research studies in organizations like Organization B to find out how requirements are managed and what the reasons are for this success.
3. The participating organizations in this research study identified various web applications they use in their project environments to handle and manage requirements management processes. The most mentioned were Pivotal Tracker, JIRA Agile, Basecamp and Trello. The researcher recommends that further research studies be carried out to find out how these web applications can be adopted and improved to provide full requirements traceability.
4. It is important to state that the research findings in this study were limited and it is recommended more cases need to be studied in order to provide a better conclusion and understanding of requirements traceability in software and IT projects.

5.5 CONCLUSIONS

In final remarks, the research carried out in this study has provided evidence of how software development and IT organizations manage and trace requirements. The evidence has shown that traceability of requirements exist but is not adequate. There is an opportunity for software development and IT organizations to start improving their products quality and increase their productivity through embracing the ideas of requirements tracing.

REFERENCES

- Abawi, K. (2008). Qualitative and quantitative research. *Reproductive health research methodology training at the ministry of public health, Afghanistan*, http://www.gfmer.ch/Medical_education_En/Afghanistan_2008/pdf/Qualitative_quantitative_research_Abawi_Afgghanistan_2008.Pdf.
- Arkley, P., & Riddle, S. (2005, August). Overcoming the traceability benefit problem. In *Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on* (pp. 385-389). IEEE.
- Atkinson, R. (1999). Project management: cost, time and quality, two best guesses and a phenomenon, it's time to accept other success criteria. *International journal of project management*, 17(6), 337-342.
- Asghar, S., & Umar, M. (2010). Requirement engineering challenges in development of software applications and selection of customer-off-the-shelf (COTS) components. *International Journal of Software Engineering*, 1(1), 32-50.
- Asuncion, H. U., François, F., & Taylor, R. N. (2007, September). An end-to-end industrial software traceability tool. In *Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering* (pp. 115-124). ACM.
- Bhattacharjee, A. (2012). *Social science research: principles, methods, and practices*.
- Bliss, L. A., & Rocco, T. S. (2013). "Mind the gap": Qualitative Researchers and Mixed Methods Research.
- Briggs, R. O., & Gruenbacher, P. (2002, January). EasyWinWin: Managing complexity in requirements negotiation with GSS. In *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on* (pp. 10-pp). IEEE.
- Boehm, B., Grunbacher, P., & Briggs, R. O. (2001). Developing groupware for requirements negotiation: lessons learned. *Software, IEEE*, 18(3), 46-55.

- Brace, W., & Cheutet, V. (2012). A framework to support requirements analysis in engineering design. *Journal of Engineering Design*, 23(12), 876-904.
- Brockington, D., & Sullivan, S. (2003). Qualitative Research. In R. Scheyvens, & D. Storey (Eds.), *Development Fieldwork*. (pp. 58-75). London, England: SAGE Publications Ltd. doi: <http://dx.doi.org.ezproxy.uct.ac.za/10.4135/9781849208864.n4>
- Bryman, A., & Burgess, B. (Eds.). (2002). *Analyzing qualitative data*. Routledge.
- Carr, J. J. (2000). Requirements engineering and management: the key to designing quality complex systems. *The TQM Magazine*, 12(6), 400-407.
- Casey, V., & Mc Caffery, F. (2013). A lightweight traceability assessment method for medical device software. *Journal of Software: Evolution and Process*, 25(4), 363-372.
- Chung, L., Nixon, B., Yu, E., & Mylopoulos, J. (2000). Non-functional requirements. *Software Engineering*.
- Cleland-Huang, J., Chang, C. K., & Christensen, M. (2003). Event-based traceability for managing evolutionary change. *Software Engineering, IEEE Transactions on*, 29(9), 796-810.
- Cleland-Huang, J., Zemont, G., & Lukasik, W. (2004, September). A heterogeneous solution for improving the return on investment of requirements traceability. In *Requirements Engineering Conference, 2004. Proceedings. 12th IEEE International* (pp. 230-239). IEEE.
- Cockburn, A., & Highsmith, J. (2001). Agile software development, the people factor. *Computer*, 34(11), 131-133.
- Corbetta, P., (2003). The Qualitative Interview. In *Social Research: Theory, Methods and Techniques*. (pp. 264-287). London: SAGE Publications Ltd. doi: <http://dx.doi.org/10.4135/9781849209922.n10>.
- Creswell, J.W. (2013) *Qualitative inquiry and research design. Choosing among five approaches*. Third ed. Los Angeles. CA: SAGE.

Cynthia G.L. Clamp, Gough, S., & Land, L. (2004). Data Collection. In *Resources for Nursing Research*. (4th, pp. 196-198). London, England: SAGE Publications, Ltd. doi: <http://dx.doi.org.ezproxy.uct.ac.za/10.4135/9780857024633.d87>.

Darke, P., Shanks, G., & Broadbent, M. (1998). Successfully completing case study research: combining rigour, relevance and pragmatism. *Information systems journal*, 8(4), 273-289.

Dömges, R., & Pohl, K. (1998). Adapting traceability environments to project-specific needs. *Communications of the ACM*, 41(12), 54-62.

Easterbrook, S., Singer, J., Storey, M. A., & Damian, D. (2008). Selecting empirical methods for software engineering research. In *Guide to advanced empirical software engineering* (pp. 285-311). Springer London.

Egyed, A., & Grunbacher, P. (2002). Automating requirements traceability: Beyond the record & replay paradigm. In *Automated Software Engineering, 2002. Proceedings. ASE 2002. 17th IEEE International Conference on* (pp. 163-171). IEEE.

Farquhar, J. D. (2012). Data collection. In *Case study research for business*. (pp. 65-84). London: SAGE Publications Ltd. doi: <http://dx.doi.org.ezproxy.uct.ac.za/10.4135/9781446287910.n6>.

Gibbert, M., Ruigrok, W., & Wicki, B. (2008). What passes as a rigorous case study?. *Strategic management journal*, 29(13), 1465-1474.

Gotel, O. C., & Finkelstein, C. W. (1994, April). An analysis of the requirements traceability problem. In *Requirements Engineering, 1994., Proceedings of the First International Conference on* (pp. 94-101). IEEE.

Gotel, O., & Finkelstein, A. (1995, March). Contribution structures [Requirements artifacts]. In *Requirements Engineering, 1995., Proceedings of the Second IEEE International Symposium on* (pp. 100-107). IEEE.

Glazer, H. Entinex, Inc., Jeff Dalton, Broadword Solutions Corporation, David Anderson, David J. Anderson & Associates, Inc. *CMMI® or Agile: Why Not Embrace Both*. CMU/SEI-2008-TN-003.

Hayes, J. H., Dekhtyar, A., & Osborne, J. (2003, September). Improving requirements tracing via information retrieval. In *Requirements Engineering Conference, 2003. Proceedings. 11th IEEE International* (pp. 138-147). IEEE.

Hayes, J. H., Dekhtyar, A., Sundaram, S. K., Holbrook, E. A., Vadlamudi, S., & April, A. (2007). REquirements TRacing On target (RETRO): improving software maintenance through traceability recovery. *Innovations in Systems and Software Engineering*, 3(3), 193-202.

Hayes, J. H., Dekhtyar, A., Sundaram, S. K., & Howard, S. (2004, September). Helping analysts trace requirements: An objective look. In *Requirements Engineering Conference, 2004. Proceedings. 12th IEEE International* (pp. 249-259). IEEE.

Heindl, M., & Biffel, S. (2005, September). A case study on value-based requirements tracing. In *Proceedings of the 10th European software engineering conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering* (pp. 60-69). ACM.

Hove, S. E., & Anda, B. (2005, September). Experiences from conducting semi-structured interviews in empirical software engineering research. In *Software Metrics, 2005. 11th IEEE International Symposium* (pp. 10-pp). IEEE.

International Council of System Engineering. 2014. *SE Tools Taxonomy - Requirements Traceability Tool*. INCOSE September 2004. Available : http://www.incose.org/productspubs/products/setools/tooltax/reqtrace_tools.html [2014, May 2].

Iqbal, M. A., Azam, F., & Qureshi, A. A. (2011). Critical Aspects of Project Management Causing Major Impacts on Software Development. *International Journal of Computer Applications*, 27.

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

Kanjanda, R. (2012). *Investigation of the Nature and Extent of Adoption and Application of Requirement Management Best Practices in Software Development Projects in South Africa*. Unpublished Master's thesis. University of Cape Town.

Kaplan, B., & Maxwell, J. A. (2005). Qualitative research methods for evaluating computer

information systems. In *Evaluating the Organizational Impact of Healthcare Information Systems* (pp. 30-55). Springer New York.

Kirova, V., Kirby, N., Kothari, D., & Childress, G. (2008). Effective requirements traceability: Models, tools, and practices. *Bell Labs Technical Journal*, 12(4), 143-157.

Flick, U., von Kardoff, E., & Steinke, I. (Eds.). (2004). *A companion to qualitative research*. Sage.

Kotonya, G., & Sommerville, I. (1996). Requirements engineering with viewpoints. *Software Engineering Journal*, 11(1), 5-18.

Lam, W., & Shankararaman, V. (1999). Requirements change: a dissection of management issues. In *EUROMICRO Conference, 1999. Proceedings. 25th* (Vol. 2, pp. 244-251). IEEE.

Lee, C., Guadagno, L., & Jia, X. (2003, October). An agile approach to capturing requirements and traceability. In *Proceedings of the 2nd International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE 2003)*. Citeseer.

Leffingwell, D., & Widrig, D. (2002). The role of requirements traceability in system development. *The Rational Edge*, 2.

Mader, P., & Egyed, A. (2012, September). Assessing the effect of requirements traceability for software maintenance. In *Software Maintenance (ICSM), 2012 28th IEEE International Conference on* (pp. 171-180). IEEE.

Mäder, P., Jones, P. L., Zhang, Y., & Cleland-Huang, J. (2013). Strategic Traceability for Safety-Critical Projects. *Software, IEEE*, 30(3), 58-66.

Malterud, K. (2001). Qualitative research: standards, challenges, and guidelines. *The lancet*, 358(9280), 483-488.

Michael D. Myers, & Avison, D. (Eds.). (2002). *Qualitative Research in Information Systems*. London, England: SAGE Publications, Ltd. doi:
<http://dx.doi.org.ezproxy.uct.ac.za/10.4135/9781849209687>

Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook*. Sage.

Myers, M. D. (1997). Qualitative research in information systems. *Management Information Systems Quarterly*, 21, 241-242.

Nair, S., de la Vara, J. L., & Sen, S. (2013). A Review of Traceability Research at the Requirements Engineering Conference. In *21st IEEE International Requirements Engineering Conference (RE'13)*.

Neto, F. M. M., & Morais, M. J. D. O. (2013). An agent-based approach for supporting the knowledge transfer in the software requirements engineering. *International Journal of Business Information Systems*, 12(1), 23-43.

Nuseibeh, B., & Easterbrook, S. (2000, May). Requirements engineering: a roadmap. In *Proceedings of the Conference on the Future of Software Engineering* (pp. 35-46). ACM.

Paetsch, F., Eberlein, A., & Maurer, F. (2003, June). Requirements engineering and agile software development. In *2012 IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises* (pp. 308-308). IEEE Computer Society.

Perry, D. E., Sim, S. E., & Easterbrook, S. M. (2004, May). Case studies for software engineers. In *Software Engineering, 2004. ICSE 2004. Proceedings. 26th International Conference on* (pp. 736-738). IEEE.

Pope, C., Ziebland, S., & Mays, N. (2000). Analysing qualitative data. *Bmj*, 320(7227), 114-116.

Project Management Institute. (2008). *A Guide to the Project Management Body of Knowledge: Newtown Square, Pennsylvania*. Project Management Institute.

Ramesh, B. (1998). Factors influencing requirements traceability practice. *Communications of the ACM*, 41(12), 37-44.

Ramesh, B., Dwiggins, D., DeVries, G., & Edwards, M. (1995). Towards requirements traceability

models. In *Systems Engineering of Computer Based Systems, 1995., Proceedings of the 1995 International Symposium and Workshop on* (pp. 229-232). IEEE.

Ramesh, B., & Edwards, M. (1993, January). Issues in the development of a requirements traceability model. In *Requirements Engineering, 1993. Proceedings of IEEE International Symposium on* (pp. 256-259). IEEE.

Ramesh, B., & Jarke, M. (2001). Toward reference models for requirements traceability. *Software Engineering, IEEE Transactions on*, 27(1), 58-93.

Rempel, P., Mader, P., & Kuschke, T. (2013, July). An empirical study on project-specific traceability strategies. In *Requirements Engineering Conference (RE), 2013 21st IEEE International* (pp. 195-204). IEEE.

Salem, A. M. (2006, March). Improving software quality through requirements traceability models. In *Computer Systems and Applications, 2006. IEEE International Conference on*. (pp. 1159-1162). IEEE.

Salem, A. M. (2006, March). Improving software quality through requirements traceability models. In *Computer Systems and Applications, 2006. IEEE International Conference on*. (pp. 1159-1162). IEEE.

Sandelowski, M. (2000). Focus on research methods-whatever happened to qualitative description?. *Research in nursing and health*, 23(4), 334-340.

Sayre, S. (Ed.). (2001). *Qualitative Methods for Marketplace Research*. Thousand Oaks, CA: SAGE Publications, Inc. doi: <http://dx.doi.org.ezproxy.uct.ac.za/10.4135/9781412985543>

Seaman, C. B. (1999). Qualitative methods in empirical studies of software engineering. *Software Engineering, IEEE Transactions on*, 25(4), 557-572.

Simons, H. (2009). Evolution and Concept of Case Study Research. In *Case Study Research in Practice*. (pp. 12-28). London: SAGE Publications, Ltd. doi: <http://dx.doi.org.ezproxy.uct.ac.za/10.4135/9781446268322.n1>.

Spanoudakis, G., Zisman, A., Pérez-Minana, E., & Krause, P. (2004). Rule-based generation of requirements traceability relations. *Journal of Systems and Software*, 72(2), 105-127.

Terry Bahill, A., & Henderson, S. J. (2005). Requirements development, verification, and validation exhibited in famous failures. *Systems Engineering*, 8(1), 1-14.

Verner, J., Cox, K., Bleistein, S., & Cerpa, N. (2007). Requirements engineering and software project success: an industrial survey in Australia and the US. *Australasian Journal of Information Systems*, 13(1), 225-238.

Wieringa, R. (1995). An introduction to requirements traceability.

Wohlin, C., Höst, M., & Henningsson, K. (2003). Empirical research methods in software engineering. In *Empirical methods and studies in software engineering* (pp. 7-23). Springer Berlin Heidelberg.

Zowghi, D., & Nurmuliani, N. (2002). A study of the impact of requirements volatility on software project performance. In *Software Engineering Conference, 2002. Ninth Asia-Pacific* (pp. 3-11). IEEE.

Appendix A

Confidentiality and consent letter

Dear Participant,

I am carrying out a research study on effective and efficient requirement traceability in the software development and information technology industry as part of my MSc Project Management studies with the University of Cape Town. The research is set to determine how *well the idea of tracing requirements is used in the aforementioned industry and what role it plays in producing quality and successful projects.*

I am kindly requesting you to participate in this research study. Participation in this research study is voluntary and you are free to reject this request. You are also free to withdrawal from participation at any stage. Participation in this research will require you to answer interview questions which may take between 30 – 60 minutes long. Your responses to the interview questions, your organization and personal information will be treated with the highest level of confidentiality. The information collected will not be disclosed to any third-party and will only be used within the confines of this research study

Your participation in this research study is highly valued and will aid greatly to the achievement of the goals and objectives of this study. It is encouraged that the parties taking part in the research study sign a confidentiality agreement that protects the interests of the participating organisation.

I, Tafadzwa Shereni, undertake to safeguard the information collected, by regarding and treating it as confidential. The information will be used in this research study only and may not be disclosed to any third-party.

I, _____ (print name), am fully aware of the aim, motivation, and purpose of this study and _____ (disagree/agree), to participate in this study.

SIGNED AT _____ ON THIS THE _____ DAY OF

PARTICIPANT: _____

PRINCIPLE RESEARCHER: _____

Email: tcshereni@gmail.com

Mobile: +27 79 127 0997

Appendix B

Cover letter

Thank you for responding to my invitation to participate in this research study. As a follow up to our earlier conversation, I am carrying out a research study on effective and efficient requirement traceability in the software development and information technology industry as part of my MSc Project Management studies with the University of Cape Town. The research is set to determine how well the idea of tracing requirements is used in the aforementioned industry and what role it plays in producing quality and successful projects.

Attached here is a consent and confidentiality letter which explains what the research is all about and how the information you provide will be treated. Also attached here are the interview questions for you to read through and familiarise with the issues that will be discussed in the interview session.

Please, advise me on suitable date and time that you would like us to come and conduct the interview. Your participation is greatly valued and am looking forward to hear from you soon.

Kind Regards,

Tafadzwa Shereni

(Principle Researcher)

Email: tcshereni@gmail.com

Mobile: +27 79 127 0997

Appendix C

Signed confidentiality and consent letter

Confidentiality and consent letter

Dear Participant,

I am carrying out a research study on effective and efficient requirement traceability in the software development and information technology industry as part of my MSc Project Management studies with the University of Cape Town. The research is set to determine how well the idea of tracing requirements is used in the aforementioned industry and what role it plays in producing quality and successful projects.

I am kindly requesting you to participate in this research study. Participation in this research study is voluntary and you are free to reject this request. You are also free to withdrawal from participation at any stage. Participation in this research will require you to answer interview questions which may take between 30 – 60 minutes long. Your responses to the interview questions, your organization and personal information will be treated with the highest level of confidentiality. The information collected will not be disclosed to any third-party and will only be used within the confines of this research study.

Your participation in this research study is highly valued and will aid greatly to the achievement of the goals and objectives of this study. It is encouraged that the parties taking part in the research study sign a confidentiality agreement that protects the interests of the participating organisation.

I, Tafadzwa Shereni, undertake to safeguard the information collected, by regarding and treating it as confidential. The information will be used in this research study only and may not be disclosed to any third-party.

I, DANIEL MAREE (print name), am fully aware of the aim, motivation, and purpose of this study and agree (disagree/agree), to participate in this study.

SIGNED AT CAPE TOWN ON THIS THE 6th DAY OF AUGUST 2014

PARTICIPANT: Musa

PRINCIPLE RESEARCHER: Shereil

Email: tcshereni@gmail.com

Mobile: +27 79 127 0997

Confidentiality and consent letter

Dear Participant,

I am carrying out a research study on effective and efficient requirement traceability in the software development and information technology industry as part of my MSc Project Management studies with the University of Cape Town. The research is set to determine how well the idea of tracing requirements is used in the aforementioned industry and what role it plays in producing quality and successful projects.

I am kindly requesting you to participate in this research study. Participation in this research study is voluntary and you are free to reject this request. You are also free to withdrawal from participation at any stage. Participation in this research will require you to answer interview questions which may take between 30 – 60 minutes long. Your responses to the interview questions, your organization and personal information will be treated with the highest level of confidentiality. The information collected will not be disclosed to any third-party and will only be used within the confines of this research study.

Your participation in this research study is highly valued and will aid greatly to the achievement of the goals and objectives of this study. It is encouraged that the parties taking part in the research study sign a confidentiality agreement that protects the interests of the participating organisation.

I, Tafadzwa Shereni, undertake to safeguard the information collected, by regarding and treating it as confidential. The information will be used in this research study only and may not be disclosed to any third-party.

I, Darren Kerr (print name), am fully aware of the aim, motivation, and purpose of this study and agree (disagree/agree), to participate in this study.

SIGNED AT Cape Town ON THIS THE 6th DAY OF August 2014

PARTICIPANT: Signed by candidate

PRINCIPLE RESEARCHER: _____

Email: tcshereni@gmail.com

Mobile: +27 79 127 0997

Confidentiality and consent letter

Dear Participant,

I am carrying out a research study on effective and efficient requirement traceability in the software development and information technology industry as part of my MSc Project Management studies with the University of Cape Town. The research is set to determine how well the idea of tracing requirements is used in the aforementioned industry and what role it plays in producing quality and successful projects.

I am kindly requesting you to participate in this research study. Participation in this research study is voluntary and you are free to reject this request. You are also free to withdrawal from participation at any stage. Participation in this research will require you to answer interview questions which may take between 30 – 60 minutes long. Your responses to the interview questions, your organization and personal information will be treated with the highest level of confidentiality. The information collected will not be disclosed to any third-party and will only be used within the confines of this research study.

Your participation in this research study is highly valued and will aid greatly to the achievement of the goals and objectives of this study. It is encouraged that the parties taking part in the research study sign a confidentiality agreement that protects the interests of the participating organisation.

I, Tafadzwa Shereni, undertake to safeguard the information collected, by regarding and treating it as confidential. The information will be used in this research study only and may not be disclosed to any third-party.

I, Willem Opperman (print name), am fully aware of the aim, motivation, and purpose of this study and agree (disagree/agree), to participate in this study.

SIGNED AT Cape Town ON THIS THE 19/08/14 DAY OF August 14.

PARTICIPANT: _

PRINCIPLE RESEARCHER: _____

Email: tcshereni@gmail.com

Mobile: +27 79 127 0997

Confidentiality and consent letter

Dear Participant,

I am carrying out a research study on effective and efficient requirement traceability in the software development and information technology industry as part of my MSc Project Management studies with the University of Cape Town. The research is set to determine how well the idea of tracing requirements is used in the aforementioned industry and what role it plays in producing quality and successful projects.

I am kindly requesting you to participate in this research study. Participation in this research study is voluntary and you are free to reject this request. You are also free to withdrawal from participation at any stage. Participation in this research will require you to answer interview questions which may take between 30 – 60 minutes long. Your responses to the interview questions, your organization and personal information will be treated with the highest level of confidentiality. The information collected will not be disclosed to any third-party and will only be used within the confines of this research study.

Your participation in this research study is highly valued and will aid greatly to the achievement of the goals and objectives of this study. It is encouraged that the parties taking part in the research study sign a confidentiality agreement that protects the interests of the participating organisation.

I, Tafadzwa Shereni, undertake to safeguard the information collected, by regarding and treating it as confidential. The information will be used in this research study only and may not be disclosed to any third-party.

I, Stuart Corbishley (print name), am fully aware of the aim, motivation, and purpose of this study and agree (disagree/agree), to participate in this study.

SIGNED AT Siyelo ON THIS THE 19th DAY OF
August

PARTICIPANT: _____

PRINCIPLE RESEARCHER: _____

Email: teshereni@gmail.com

Mobile: +27 79 127 0997

Confidentiality and consent letter

Dear Participant,

I am carrying out a research study on effective and efficient requirement traceability in the software development and information technology industry as part of my MSc Project Management studies with the University of Cape Town. The research is set to determine how well the idea of tracing requirements is used in the aforementioned industry and what role it plays in producing quality and successful projects.

I am kindly requesting you to participate in this research study. Participation in this research study is voluntary and you are free to reject this request. You are also free to withdraw from participation at any stage. Participation in this research will require you to answer interview questions which may take between 30 – 60 minutes long. Your responses to the interview questions, your organization and personal information will be treated with the highest level of confidentiality. The information collected will not be disclosed to any third-party and will only be used within the confines of this research study.

Your participation in this research study is highly valued and will aid greatly to the achievement of the goals and objectives of this study. It is encouraged that the parties taking part in the research study sign a confidentiality agreement that protects the interests of the participating organisation.

I, Tafadzwa Shereni, undertake to safeguard the information collected, by regarding and treating it as confidential. The information will be used in this research study only and may not be disclosed to any third-party.

I, Aslam Bagus (print name), am fully aware of the aim, motivation, and purpose of this study and agree (disagree/agree), to participate in this study.

SIGNED AT CPT ON THIS THE 27th DAY OF August 14.

PARTICIPANT: _____

PRINCIPLE RESEARCHER: _____

Email: tcshereni@gmail.com

Mobile: -27 79 127 0997

Appendix D

Research interview questions

Date	
Interviewee	
Role in the Organisation	

Requirements

1. Does your organization have a formal requirements management process which helps in gathering and storing requirements?
2. Does your organisation maintain any sort of documents pertaining to requirements? If so, please elaborate.
3. Is your organization involved in any requirements process like elicitation, analysis, communication, validation and management? If so, how important do you think this involvement is to the overall project success?

Requirements Traceability

4. What do you understand by the term “requirements traceability”?
5. Requirement traceability can be implemented through documentation or design artefacts. How does your organisation implement it?
6. *Does your organisation plan requirements traceability? If so, in what way and at what stage of the project phase?*
7. Does your projects environment use any automation models and tools to capture requirements traces? If so, how effective and efficient are they?
8. In what kind of database are the traces stored?
9. Is your organization involved in decision making on how, when, and who should establish and maintain the database where these traces are stored?
10. How easily accessible is the information stored in the database?
11. How valuable is requirements traceability to your organization?
12. How important is it to you?
13. How much does your clients value requirement traceability and how important do you think it is to them?

14. How cost effective is requirement traceability to your organization?

Requirements to product mapping

15. Do you have test cases in your project environment?

16. How easy is to find information about a particular requirement?

17. In your development environment does each module, class or function identify the requirements it fulfils? If so, how?

Change management

18. Does your organisation have a formal process to handle changes to requirements?

19. How do you ensure effective change management?

20. Who authorizes changes to requirements?

21. Do you have any version control mechanisms? If so, does each version control change identify the requirement(s) being addressed?

Conclusion

22. How effective and efficient do you think are the processes used in your organisation in requirements management and requirements traceability?

Appendix E

Research interview responses

Organization A

The participant was the Product Manager, who once served as the Project Manager for 3 years before his current role.

Requirements and requirements management processes

- Formal requirements management processes exists but the processes are not strictly followed. There are guidelines which explain how things should be done and templates exists that help the people in all projects.
- Requirements documentation are maintained in the form of screen mark-ups, design artefacts, wire-frames, minutes of meetings where requirements are clarified, requirements documents where requirements are captured, issue documents from clients, and traceability matrix. Traceability matrix is not always employed but is used dependent on a project.
- The organisation is totally involved in all the requirements processes and views the involvement as critically important for them in delivering a successful project.
- Often requirements are fairly big in volume and client organizations do not understand their own requirements in most cases, so the organization decided to be involved right up front when the requirements are being gathered and throughout the project life-cycle.

Requirements traceability

- Requirements traceability is seen as delivering what the client has asked for, all the way through implementation, tested and approved.
- Requirements traceability is implemented through documents such as traceability matrix and design artefacts, wire-frames and screen mark-ups.
- Requirement traceability is not always planned. There is a minimal planning involved. In some cases it is a deliverable specified by the client, in such a case planning is done. Certain projects are done in agile development environment which requires user stories and the traceability matrix which are redefined and refined as the project evolves.
- No to automation, no to tools. Tools that are used are for managing the project processes and produce documents like traceability matrix which have links and references pointing to different locations.

- JIRA – Agile is used for most of the projects and it helps in planning and executing scrum activities.
- They use excel spreadsheets
- The organisation is expected to produce the traceability of things and the client do not care much about the format and how the traces are stored.
- Quite easily accessible but spreadsheets are bit difficult if it's quite big.
- To the organisation it's not valuable but to the client. It's mainly a deliverable.
- It's valuable when you have a third-party or a very distance stakeholder to deal it. Scrum requires to log everything and it prioritise it, all what done and what's not done is known, so traceability is not much of an importance unless somebody asks for it.
- Client think it's quite important and use it to measure if they got what they ask for.
- Costs are incurred since documents like traceability matrix requires a team member or two to maintain.

Requirements to product mapping

- Test cases exists. Test case scenarios are written.
- JIRA – has a fairly flexible query language to search for items. Requirements are categorized by theme and so on for easy searches.
- It sort of does. No comments in code. The naming convention explains. Acceptance test scenarios are written upfront that are automated. The scenarios are done to address specific requirements. Scenarios are in clear English statements that relates to requirements

Requirements change management

- There is a formal process. Depending on the kind of change you dealing with, the severity of the formality change. For something that affects time and budget requires sign off of documents. Some they just do if trust exists with the client.
- Assessment of how risk the change is and that determines how formal the process has to be.
- Authorization depends on client environment and the team.
- Version control, source control on all projects. Depending on projects a version scheme can be used.

Conclusion

- Requirements management is fairly effective.
- Requirements traceability, is not always effective since they not always do it.

Organization B

The participant was the Project Manager who has been working for the companies for 2 and half years.

Requirements and requirements management processes

- The organization has no formal processes or approaches that are followed on requirements.
- The organization initially organize a series of meetings with client, The people involved in these meeting include project leaders, developers, designers and stakeholder representative from the client side.
- In these meetings the client provides all the requirement's that are in-line with his/her needs. These high level requirements are captured in and saved in a spreadsheet on Google Docs.
- Depending on clients' needs, a specification document is created stored in Google docs and it is shared with all concerned parties.
- Once the specification is produced, the clients signs it off.
- When all stakeholders are happy and are agreed on the requirements, low level requirements and tasks are created and stored in an online application called Trello.
- The organisation is involved in all the requirements processes despite the fact that they do not have a formal process. The organization is involved from the beginning and are hands on with all the requirements processes.
- The organization's involvement in requirement processes help them to understand what exactly their clients need. In most cases clients do understand what they want but they do not know the implications of what their wants when it comes to the real world.

Requirements traceability

- Requirement traceability is all about being able to go back to the beginning where project requirements are first made into specification. Often requirements change and we need the ability to go back to the original idea.
- Traceability is not really planned but it is vitally important to start thinking about it right at the beginning of a project.
- No automation tools or models used. Trello is a web application accessible online.
- Requirements are store in Google docs, Trello.
- Google docs provides historical records, Trello has an archiving facility. Both allows for easy storage of requirements traces.

- The organisation handles all the requirements and are responsible for maintaining and managing the traces.
- Trello and Google Docs are easily accessible, once a document or requirement is shared it can be seen by all people in the group.
- Traceability is is very vital for the organisation because often things go wrong and they need ability to trace where the problem emanates from. Often time problems come from requirement and how there are implemented.
- Design keeps changing and traceability helps especially when issues arise with clients.
- It makes life easy. Clients are not very technical and their minds keep on changing, so traceability extremely valuable.
- Extremely important to them but they don't know and understand it.
- Traceability can boost productivity.

Requirements to product mapping

- Test cases exists, the organization runs a lot of tests, from small code module, functions and to classes. The development environment has test suites for example PHP Unit, selenium for web application testing.
- Module, function and class definitions are normally done in easy to understand language, which reflect the functions or requirement being addressed. Comments and to-dos are used in code to mark what need to be done and when it should be done. This aids a lot in system maintenance.
- It's very easy to find information on any requirement because everything is track-able in Google Docs and Trello.
- Help and installation files are also created.

Requirements change management

- No formal process of change management exists but change is handled mostly in meetings that are conducted twice a day. The first meeting is a capture meeting to review what has been done so far and what need to be done for the day. The second is a progress review of what has been done so far and what need to be done in the future. If any changes to requirements are discussed in these meeting and Google Docs and Trello are updated accordingly.
- Changes to requirements are discussed with all the stakeholders and whatever is agreed

documented and implemented.

- The client normally authorizes all the requirements but the project leader can have influence on what the requirements should be like. He/She can decide what to implement and not implement then clarify that to the client. The client is made to see alternative ways of achieving particular needs.
- Version control exists in all documents. Google Docs provides a history of document changes. Trello has an archiving facility which allows the tracking of all changes. The development environments have tracking abilities like git and source control.

Conclusion

- The nature of project management is forever changing, the tools used by the organization are useful and serve the purpose. The processes are effective and are helping to increase productivity and shorten the time a project is delivered to the client.

Organization C

The participant is a senior development supervisor, who has been with the company for more than six years.

Requirements and requirements management processes

- The organisation is focused on agile methodologies specifically SCRUM.
- SCRUM is not very strict on requirements or documents traceability and focuses on a collaborative approach with various stakeholders as well as allowing requirements not to be too prescriptive upfront but rather allowing them to develop and change through the duration of a project.
- Requirements are agreed upon in a mutual agreement basis.
- The requirements are kept in a project or product backlog, which is a documented used in agile development environments.
- The organization is involved in all activities and processes in their projects. Requirements are agreed upon in a collaborative approach, with the project sponsor playing a major role in defining the requirements.

Requirements traceability

- The organization understands that SCRUM deals with frequently changing requirements, and is not strict on documents so tracing requirements is not a major issue.
- The project or product backlogs are updated frequently, so they act as project documents and all the changes are noted and made known to all stakeholders.
- Use of user stories
- The organization uses VersionOne and JIRA for managing all SCRUM activities.
- Everything is stored in backlog and excel sheets.
- The clients decide on requirements but what is implemented is everything agreed in a sprint. Everything is mutually agreed on.
- Information stored in JIRA and VersionOne is easily accessible.
- Requirements tracing in the traditional way of matrices and requirements documents is no longer as important as it used to be. The organization concentrate on managing the people over the processes.

- Tracing in the modern methods especially agile development, protects the organization from unforeseen problems. The client has to sign off every user story and that acts as a contract.
- Traceability is not important to clients, what they want to see is a system that works as they deem necessary.
- The organization does not spend much money on tracing requirements. The tools used are yearly subscriptions that are affordable.

Requirements to product mapping

- Test cases exists
- Information on requirements is easy. The user stories tell everything about what is required.
- In most cases the requirements are reflected in the way modules and functions are defined. The naming used is simple English which reflects what the class or function do.

Requirements change management

- No properly defined processes exists to handle changes to requirements
- The client has the power to change requirements but they communicate and everything is dealt with in SCRUM sessions.
- The project owner and all stakeholder can authorized.
- Use of SVN and git depending on project.

Conclusion

- The processes used are useful and efficient. Agile development is suitable for rapidly changing requirements and they believe in what it can do and achieve.
- Clients are happy too and they trust the processes and methods used.

Organization D

The participant was a senior developer who has worked for the company for more than 7 years.

Requirements and requirements management processes

- The organization has as formal process of management requirements and follow the agile manifesto as much as it makes sense to them and their clients.
- Requirements are gathered usually in a meeting held every week or two depending on the sprint cycle. They keep on gathering the requirements until a point where they know that they have enough for the next obligation.
- They do not want to plan too far upfront because things always change.
- They organization uses pivotal tracker to manage and store the requirements. They do not have a formal documents like traceability matrix. The use stories on pivotal tracker to communicate within the team and with the clients. They use white board and take pictures.
- The organization is involved in absolutely all the processes that are to deal with requirements. It is important to actually sit through all the processes because the architectural decisions of the product/project are influenced by every word that comes from the mouth of the client. Spoken word can help you gauge the level of importance a certain requirement is. Participation is of paramount importance.

Requirements traceability

- Management of a requirement from project inception to the final product
- User stories, design artefacts
- No planning when it comes to traceability. Signoff is done on story cards. The clients are however held responsible for the stories. They accept what is written. Story cards used for signoff.
- No automation, other than pivotal tracker.
- Pivotal tracker gives you Charts, burn downs etc.
- Organization decides what it need to use.
- Backlog is used to track things that needs to be done. Backlog feeds into the next SCRUM sprint.
- Pivotal tracker has a clouds service, iphone apps
- Accountability

- Helps the organization in tracking team performance and what each team member is doing
- Obligation for doing something
- It can be a deliverable
- It builds trust between client and the organization

Requirements to product mapping

- Test cases exist, behavioural driven IT, units and functional test. Test everything
- Mapping business requirement to code. Comments are about intention. Link stories to commit messages,
- Comments are traceable.

Requirements change management

- No formal processes exist to handle change requirements.
- Changes can be managed and authorized by any team member.
- Clients also decide on what has to be changed
- Git, GitHub, and source control exist, CI, CD, integration server

Conclusion

- They are pretty effective and efficient. Educating clients is the biggest stop gap not necessarily the tools. The human element of managing expectations is far more important than managing processes. Agile prescribes people and communication over tools and processes. It builds trust and clients can understand how and what you can offer them. The tools and processes are just a means to remembering things.

Organization E

The participant was the application development manager who has been with the company for 6 years.

Requirements and requirements management processes

- Most requirements management activities are conducted in an informal manner. However due to requirements inconsistencies from most clients the organization has some processes and templates in place to manage unexpected changes to requirements
- Clients always change their minds and without defined processes or organization principles, requirements are difficult to manage.
- Requirements are gathered in meetings and reviews that are done twice a week. Monday meetings are to review what has been done in the previous week and the way forward and Friday meeting are for reviewing what has been in the week.
- The organisation uses Basecamp, an online application as their tool for project management. All the requirements, design artefacts, user story and test cases are managed with this software. Manual documents such as traceability matrix are used depending on the needs of the project.
- To guard against unexpected changes from the clients the organisation is involved in every requirement management processes. In most cases clients suggest requirements that are not feasible to implement so the organisation gives direction and help the client to make decisions on requirements.

Requirements traceability

- Requirements traceability is achievable through both documents and design artefacts. Basecamp allows you to upload all sorts of documents, spreadsheets, portable documents format, images of user stories and use cases.
- The organization does not plan requirements traceability, but more often development teams are enforced to do through collaborations and interaction through Basecamp.
- No automation models are used. Basecamp and manually produced documents are the only means.
- All the requirement traces are kept in Basecamp and any other possible documents like spreadsheets. Basecamp gives the ability to see historical data, and track how a particular requirement was handled.

- The organisation decides how the requirements are managed and stored. Clients are advised on how the requirements are being kept and these environment can be shared with them upon request or mutual agreement.
- Clients can use Basecamp to comment and provide more information on requirement or tasks.
- Information is quite accessible, Basecamp is accessible from anywhere. It has a search box that enable one to search on particular words. All items with that word or phrase will be displayed and are linked and clickable.
- It is very important, It helps to track all things in a project
- Most clients value it and they want it as a deliverable.
- Basecamp is a subscription software which is quite affordable, there is no need to employ someone to manage your requirements.

Requirements to product mapping

- Development environment has test cases, the framework used for development allows the development team to create unit and functional tests.
- It is difficult to show or reflect requirements in code. It will be applicable if a system is very small but when a system get bigger it's difficult to manage.

Requirements change management

- Any changes to requirements are mutually agreed upon with the clients in the meeting that are done.
- No one changes a requirement without the other party knowing, every change should be communicated and noted accordingly. That's how formal the process is.
- Changes to requirement are agreed upon by both parties but the client has the final say.
- The organization use git and trac for version control.

Conclusion

- Basecamp and any other processes we use to manage and trace requirements are very effective and efficient, although we always look for better online tools that make the processes easier.

Appendix F

Ethics clearance

EBE Faculty: Assessment of Ethics in Research Projects

Any person planning to undertake research in the Faculty of Engineering and the Built Environment at the University of Cape Town is required to complete this form before collecting or analysing data. When completed it should be submitted to the supervisor (where applicable) and from there to the Head of Department. If any of the questions below have been answered YES, and the applicant is NOT a fourth year student, the Head should forward this form for approval by the Faculty CIR committee: submit to Ms Zulphia Geyer (Zulphia.Geyer@uct.ac.za, Chem Eng Building, Ph 021 850 4791). **NB: A copy of this signed form must be included with the thesis/dissertation/report when it is submitted for examination**

This form must only be completed once the most recent revision EBE EIR Handbook has been read.

Name of Principal Researcher/Student: Tafadzwa Shereni

Department: Construction Economics and Management

Preferred email address of the applicant: icshereni@gmail.com

If a Student: Degree: MSc Project Management Supervisor: Ian Jay

If a Research Contract indicate source of funding/sponsorship:

Research Project Title: *Effective And Efficient Requirement Traceability In the Software Development And Information Technology Industry.*

Overview of ethics issues in your research project:

Question 1: Is there a possibility that your research could cause harm to a third party (i.e. a person not involved in your project)?	YES	NO <input checked="" type="checkbox"/>
Question 2: Is your research making use of human subjects as sources of data? If your answer is YES, please complete Addendum 2.	YES <input checked="" type="checkbox"/>	NO
Question 3: Does your research involve the participation of or provision of services to communities? If your answer is YES, please complete Addendum 3.	YES	NO <input checked="" type="checkbox"/>
Question 4: If your research is sponsored, is there any potential for conflicts of interest? If your answer is YES, please complete Addendum 4.	YES	NO <input checked="" type="checkbox"/>

If you have answered YES to any of the above questions, please append a copy of your research proposal, as well as any interview schedules or questionnaires (Addendum 1) and please complete further addenda as appropriate. Ensure that you refer to the EIR Handbook to assist you in completing the documentation requirements for this form.

I hereby undertake to carry out my research in such a way that

- there is no apparent legal objection to the nature or the method of research; and
- the research will not compromise staff or students or the other responsibilities of the University;
- the stated objective will be achieved, and the findings will have a high degree of validity;
- limitations and alternative interpretations will be considered;
- the findings could be subject to peer review and publicly available; and
- I will comply with the conventions of copyright and avoid any practice that would constitute plagiarism.

Signed by:

Principal Researcher/Student:	Full name and signature Tafadzwa Shereni	Date 04 July 2014
-------------------------------	---	----------------------

This application is approved by:
Supervisor (if applicable):

Ian Jay

HOD (or delegated nominee):
Final authority for all assessments with NO to all questions and for all undergraduate research.
Chair, Faculty EIR Committee
For applicants other than undergraduate students who have answered YES to any of the above questions.

13/08/2014