QUALITY MANAGEMENT CHALLENGES IN ITERATIVE SOFTWARE PRODUCT DEVELOPMENT OF A SELECTED SOFTWARE DEVELOPMENT ORGANISATION IN CAPE TOWN, SOUTH AFRICA

by

ENCILIAH CHIPUNZA
(216274001)

Thesis submitted in partial fulfilment of the requirements for the degree Master of Technology: **Business Information Systems**

in the Faculty of Business

at the Cape Peninsula University of Technology

Supervisor: Dr Michael Twum-Darko

Cape Town Campus

August 2018

DECLARATION

I, Enciliah Chipunza, student number 216274001, declare that the contents of this thesis represent my own unaided work, and that the thesis has not previously been submitted for academic examination towards any qualification. Furthermore, it represents my own opinions and not necessarily those of the Cape Peninsula University of Technology.

SIGNATURE

16 August 2018

DATE

ABSTRACT

Many software organisations using iterative software development approach use practices that relate to quality management. However, the quality management process has been inadequate. Despite many research studies conducted on quality management in iterative software product development none have adequately addressed the challenges and mitigation techniques to have an adequate process that leads to a quality software product.

The objective of this study was to determine factors that affect the quality management process in iterative software development. The research followed a qualitative approach, a case of software organisation SasTech in Cape Town, South Africa. 22 interviews were conducted on three roles actively involved in the software product development process. These are product management, quality assurance and software developers. Themes were drawn from results and were tabulated. The duality of technology theory was used as a theoretical lens to data analysis.

Several factors were identified to influence the software quality management process. These include planning, documentation, process ownership, technologies, testing, timelines and management support.

Through the general proposed framework, facilities (human resources and technologies), interpretive schemes (architecture) and norms (practices) of software quality management can be institutionalised leading to adequate and effective quality management in iterative development for SasTech as well as other organisations in the same industry.

Key words: Quality management, iterative software development, structuration theory, duality of technology

ACKNOWLEDGEMENTS

I wish to thank:

- My God, and Christ for giving me the courage and strength.
- My supervisor, Dr Michael Twum-Darko for walking with me every step of the way to completing this study.
- My family for understanding and their encouragement.
- Ms Patience Mpofu for allowing me to conduct the research at Seudobit.
- My friends Quincy, Emma and Lucky for their time and moral support.
- All the participants who were willing to assist me despite their busy schedules.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

GLOSSARY

Terms/Acronyms/Abbreviations	Definition/ Explanation
AUP	Agile Unified Process
CMM	Capability Maturity Model
DOS	Duality of Structure
DOT	Duality of Technology
ISD	Iterative Software Development
IS	Information Systems
ISO	International Organisation for Standardisation
MVP	Minimal Viable Product
QA	Quality Assurance
QM	Quality Management
SDLC	Software Development Life Cycle
ST	Structuration Theory
UAT	User Acceptance Testing
XP	Extreme Programming

CHAPTER ONE

INTRODUCTION AND BACKGROUND OF STUDY

1.1 Introduction:

Many software organisations using iterative software development approach use practices that relate to quality management. However, the quality management process by some of these organisations, particularly in Africa has been inadequate to ensure that the released software products are free from errors or bugs. In this research, Software organisations are companies which focus on the creation of innovative high-tech products. Although many software products are meant to make us more effective and efficient in many of our daily activities or operations, they are somehow disrupting our lives and lifestyles. As such they have become very important and even critical in some industries such as health and aircraft (Hinchey & Sterritt, 2006). In the software development industries, it is critical that before the product is released to customers, that it is effectively working with no errors. Some popular software organisations in today's world include Facebook, Google, LinkedIn, Instagram, Apple, Uber, Microsoft and many others. These have been hugely adopted by many people because they make their lives easier. Businesses across all industries make use of software products to optimise their processes and to function efficiently and effectively. As technology is rapidly changing so are customer needs, putting pressure on software organisations to deliver products faster while meeting customer expectations. Customers have the power of choice such that if there is any fault or missing functionality in a software product they can switch to another competing product that can meet their needs. Many disruptive "start-ups" are emerging each year making the environment competitive, concurrently other software organisations are failing to meet the market needs.

Development of software products is also becoming more and more complex because of many methodologies and technologies being developed. It is thus important that an organisation choose tools that works for it. According to Sommerville (2011), the development process which an organisation adopts has influence on the quality of the software product. Many organisations have moved from traditional processes such as "waterfall" to iterative software development (ISD) processes (Hashmi & Baik, 2007) also called ''agile.'' The most common two approaches are Extreme Programming (XP) and Scrum which many studies have claimed are more suitable for this kind of environment. The scope of this study focuses on these two approaches to iterative software development.

Although many studies have claimed that adopting an iterative development process helps in improving the quality of the product and is more suitable for complex situations, it is therefore, important to understand which aspects of quality is improved. A quality product must be of integrity, usable, reliable, accurate, efficient, maintainable, testable, flexible and reusable (Fitzpatrick, 1996).

Fitzpatrick, Smith and O'Shea (2004) defined software quality as an excellent measure to which aspects of the product and process are perceived throughout the software product life-cycle. Lewis (2016:637) defined the quality management process as the *"execution of processes and procedures that ensures quality as an output from the development process*." Quality management and improvement in ISD process, complimented by better software development techniques, tools and programming languages have led to better quality software products (Sommerville, 2011). Also, the skills and expertise of the people developing the software and the general social system play a huge part in having a quality delivery process.

According to Gill (2005) if the quality management process is inadequate, the product will fail. The key components of quality management in software development identified by Sommerville (2011) are quality planning, quality assurance and quality control. If an organisation has put in place these practices, it is highly likely the organisation will produce a quality software product. In this research, it has been argued that despite many research studies conducted on quality management in iterative software product development none have adequately addressed the challenges and mitigation techniques to have an adequate process that leads to a quality software product. A holistic approach to creating a working quality management system has become urgent, one that can be customised for each organisation. A literature review has proven that standards like ISO 9001 are not applicable to all organisations but can be used as a framework for creating a quality management system.

This case study employed a qualitative method for data collection and analysis derived from semi-structured interviews. Duality of technology model of structuration theory was used as a window through which guidelines for institutionalisation of quality management in iterative software development were derived. The intended outcome of this study was a general framework for improving quality management process in iterative and incremental software product development.

1.2 Rationale of the Study

1.2.1 Background

Software development organisations have been emerging in Africa and in particular, South Africa. One would almost say Cape Town is the Silicon Valley of Africa and hence a software development organisation was identified and selected as the Case Study for this research. Top software organisations include both consumer and business products covering all industrial sectors. The vision of these growing software development houses is to advance the development and use of technology in South Africa. These organisations are attracting investments and interest from both the government of South Africa, private sector and foreign investors.

Many of these organisations have adopted iterative software product development. Iterative software development gives an organisation flexibility to be selective on principles and practices based on the type of product and organisational culture and this can be problematic when it concerns quality management practices (Soundararajan, Arthur & Balci, 2012).

Marques, Costa, Silva and Gonçalves (2017) noted the lack of appropriate processes as a major challenge leading to software failures. The authors identified the issues of inappropriateness of the process and went on to discover that even when challenges are identifies orgnaisations often do not do anything about it leading to their failure.

Acording to a study done in 2013, 61.5% of all large organsations were challenged compared to 46.7% for medium organisations and 50.4% for small organisations (Rajkumar *&* Alagarsamy, 2013) as illustrated in figure 1.1 below.

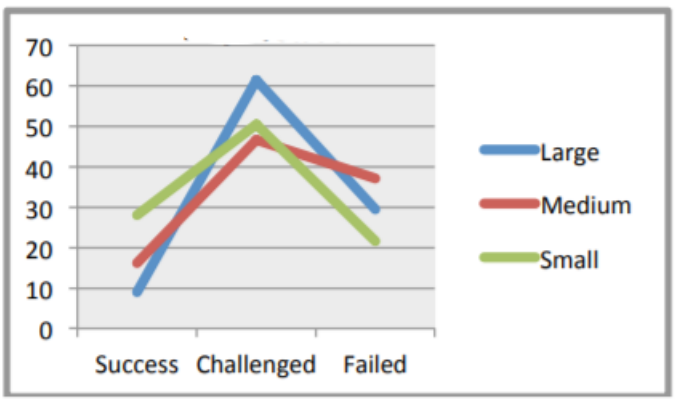


Figure 1.1: Software failure statistics

(Adapted from Rajkumar & Alagarsamy, 2013)

The Cape Town based software development organisation (SasTech - fictious name) offers software as a service to its customers as a good representation of an African grown software development organisation. It has developed many virtual products for both businesses and consumers. These are accessed via mobile apps (Android and iOS) and websites. The organisation has a national customer base and has over 100 employees, however only 24 employees are involved in the software product development process. The employees involved are organised into product management team reporting to the chief executive officer, quality assurance team currently reporting to the development manager and software development team reporting to the development manager. The product team is based in Cape Town.

The organisation has adopted an iterative software development approach. However, there are no proper processes in place that ensure quality is met throughout the whole software development life cycle (SDLC) for each iteration. There is no continuous quality delivery improvement process. The organisation's approach to product delivery has caused some of its products to stagnate whilst competing products are improving in both functionality and quality.

It has seen significant revenue losses and, so many customers have churned taking their business elsewhere over the last couple of years. The main reason is that some of its software products are filled with defects making it difficult for customers to complete certain tasks. There is a massive backlog of defects logged on their project management tool. The organisation uses Trello to manage its projects. However, with each new iteration new defects emerge. The question becomes how this can happen? What kind of processes are in place? How effective are these processes? What can be done to prevent this from happening? It becomes a dilemma for the organisation since it will need to make decision on whether to focus on new features while defects build up or fix the issues while lagging on new requirements.

Although this study will focus on SasTech processes, Sommerville (2011) note that the need for an effective quality management process is universal. A case study is suitable for this research since it allows a better study of complex issues and enables the researcher to explore data in a specific situation. The case study approach beyond qualitative study allows researchers to tap into the unknown and advance knowledge in known areas as well (Starman, 2013).

In summary, this study aimed to explore quality management factors in the software development process at a selected organisation in Cape Town, South Africa. A lot of similar studies have taken place outside South Africa.

Hence, the need for a local assessment of quality management practices to identify challenges and possible solutions to mitigate those challenges to move the local industry forward. The outcome of this study is also beneficial to the selected organisation as it will help it improve its business as well as other organisations in similar situations.

1.2.2 Problem statement

Ineffective quality management process in iterative software development in most African Countries has made many software products fail to meet customer expectations. Drawing from Giardino, Paternoster, Unterkalmsteiner, Gorschek and Abrahamsson (2016) it is arguable that many organisations especially start-ups face an uncertain market with evolving customer requirements and as such are finding it challenging to follow proper software processes.

Ineffective quality management may translate into lost business opportunities or even failed businesses (Royce, Bittner & Perrow, 2009). Therefore, it is vital to have an aggressive process in place to mitigate this risk. The issue of quality in software product development process is arguably still a prime concern for most organisations. Previous studies have focused more on traditional approaches (for example, 'waterfall' methodology) to software product development. In iterative software product development, there is an assumption that there is better quality because of the very nature of the development process. Agility in software development has been recommended from studies in the 1970s. However, adoption has been slow and many studies there-after arguing there is still a problem. Miller (2013) state the reason why some products fail is because of inexperience of the use of ISD method, the organisation's philosophy or culture.

Adopting Orlikowski's DoT model analogy, software quality management process as structure like new technology introduction is faced with acceptance and resistance. Employees will alter practices based on their knowledge, interpretation and relevance.

The organisation under study has many projects with different teams working on them to deliver software products. There is no standardised quality management process or best practices and standard procedures in place to determine how software products are to be delivered.

In summary, the research problem was to investigate the inadequate and ineffective quality management in the iterative software product delivery process which leads to organisational losses and lack of growth. This was done through the lenses of duality of technology of structuration theory and ISO 9001 quality management system.

1.3 Research aim and objectives

The aim of this study was to explore the factors influencing the ineffective quality management in iterative software development process using a software development organisation in Cape Town, South Africa as a case study. To address the aim, the main objective was:

> "*to investigate the factors influencing quality management process on the development of quality software products*."

To meet this main objective, the following subordinated objectives were found to be necessary:

(a) To determine the factors influencing quality management process of software product development;
(b) To investigate the sociotechnical processes embedded in the quality management process of software product development;
(c) To investigate what contributes to the ineffectiveness of software development processes; and
(d) To propose a normative framework to guide quality management in software product development to meet customer expectations.

1.4 Research questions

To address the above-mentioned objectives in relation to the problem statement, the main research question was:

> *"What are the factors influencing quality management process of software product development?"*

To tease out the main research question, the following sub-questions were proposed:

(a) How is quality management process in software product development influenced?
(b) How are socio-technical processes embedded in the quality management process of software product development?

(c) Why are software development processes ineffective?
(d) How can quality management in software product development be guided to meet customer expectations?

1.5 Overview of literature review

According to Schon, Thomaschewski and Escalona (2017) technology is moving rapidly and the demands of customers are rising. Customers demand high quality software products to be delivered continuously on time. Changing requirements put pressure on organisations (Chevers & Grant, 2017). Popovic (2015) found that many organisations are now adopting iterative and incremental also known as "Agile" software development processes and are moving away from traditional "waterfall" approaches.

One of the major weaknesses of the waterfall approach as stated by Mohammed, Munassar and Govardhan (2010) is that it does not meet the exact requirements of customers because these change over time. There is too much administration involved which could be costly and the product could have serious defects. To add on, customer input is most often received late because it is required in the last stages of development and this causes the development cycle to drag on as the input is implemented. The waterfall approach will most likely cause a project to go over budget. These disadvantages are claimed to be advantages for ISD approaches by several studies such as (Kruchten, 2008; Karambir & Sharma, 2016; West & Grant, 2010). They argue that an iterative approach brings higher quality products in shorter periods of time that satisfy the needs of customers.

However, the use of an ISD process is still risky, complex and unique. Determining the necessary process that will produce a quality product is not easy (Borque & Fairley, 2014). There is need to use common controls and processes that mitigate risk and avoid repeating the same mistakes. The process requires experience and skill to be effective (Mohammed *et al.*, 2010) and it needs to be adjusted to suit a situation to meet the quality required (Huo, Verner, Zhu & Babar, 2004). Approaches such as XP and Scrum are difficult to manage if it is a large or complex project and require lots of coordination. According to Flora and Chande (2014), Scrum fails on the issue of criticality and does not offer guidance on meeting quality requirement and a team will have to adopt another approach for it to be effective (Khalane & Tanner, 2013). Software development teams do not want to focus too much on process as it is perceived to reduce productivity (Clutterbuck, Rowlands & Seamons, 2008).

Sommerville (2011) illustrated how quality management could be used as an independent check in the software development process. The quality management process is there to

ensure that project deliverables are in line with organisational standards and sprint objectives as seen in figure 1.2 below:

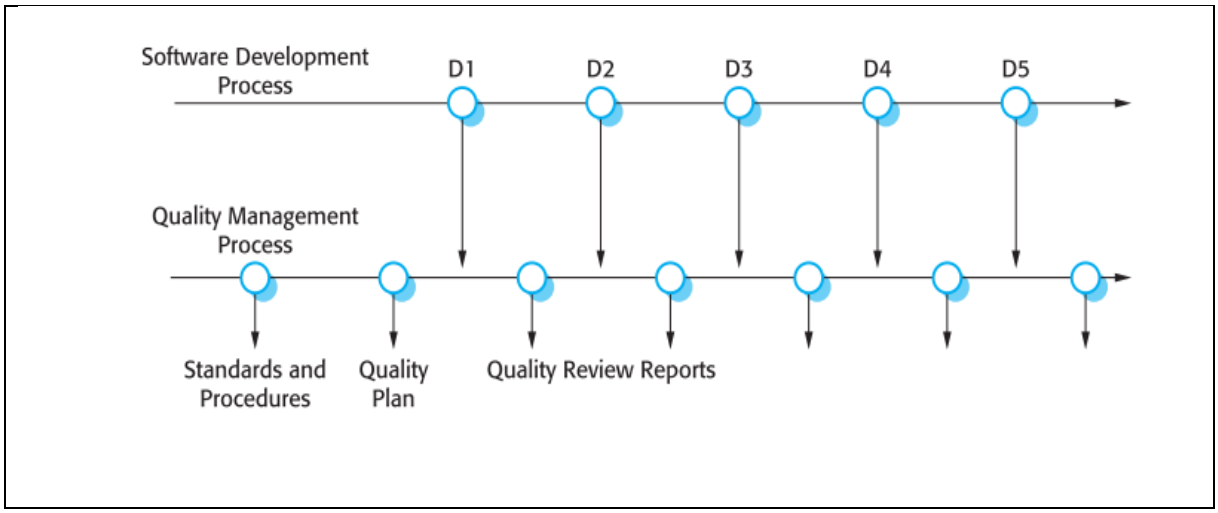


Figure 1.2: Software development and quality management

(Adapted from Sommerville, 2011)

However, the study did not explain clearly the interaction between each development process in the product life-cycle and quality management. Quality management process and improvement can lead to less dense defects in the software product developed (Sommerville, 2011). Following the quality management components, it ensures that quality issues are identified and resolved quickly (Jamsutkar, Patil & Chawan, 2012). Effective quality management may result in significant profits, reduction in inconsistencies, error-free software products and reduced costs (Khan, Keung, Niazi, Hussain & Ahmad 2017).

Osmundson, Michael, Machniak and Grossman (2003) highlighted how software organisations have constantly been attempting to improve the software development process. There are models that have been proposed such as CMM (Capability Maturity Model) and ISO (International Organisation for Standardisation) which identify key practices required to improve organisations' software development processes.

In this study, the ISO 9001 standard was analysed. ISO 9001 is the set of international standards organisations can use to develop a quality management process. It provides a guideline for developing software standards and outlines quality processes (Sommerville, 2010). "*The ISO 9001 approach posits an organisational-level quality management process paired with project-level quality assurance planning to achieve the organisational goals*" (Bourque & Fairley, 2014). According to Sommerville (2010) the core product deliverables of ISO 9001 are business acquisition, design and development, test, production and delivery and

service and support. However, some authors have claimed that these models are associated with rigidity and a lot of bureaucracy. Hence unsuitable for iterative software development processes. However, setting software product development standards assist in bringing consistency among software development teams, proposing best practices for an organisation and defining what quality is (Sommerville, 2011).

A lot of studies have criticised international standards, but some believe it is a good start when an organisation wants to start or improve its processes. Zope, Nori, Kumar, Lokku, Natarajan & Nistala (2016) argue that the practices provide guidance but do not include all the stages of the SDLC so that people playing different roles in the process are aware of their contributions to software quality and assuring it through related activities and design. Testing only validates if the products meet customer expectation for functionality and performance before deployment. According to Aregbesola, Akinkunmi and Akinola (2011) testing alone is not enough to meet business expectations and quality management.

Effective quality management may result in significant profits, reduction in inconsistencies, error-free software products and reduced costs (Khan *et al*., 2017). Zimon and Malindzak (2017) argue that this is time consuming and costly. Although the steps to effective quality management may be known, Mateen, Jahanzaib and Iqbal (2017) argued that most organisations do not properly follow the best practices and the process is not properly owned.

Abrahamsson, Salo, Ronkainen and Warsta (2017) regarding software developers stated that sometimes they lose track of quality fluctuations within the software product being developed. To achieve excellence, it is important to note that the body of knowledge needs new techniques on quality management approaches to iterative software development (Alhassan, Alzahrani & AbdulAziz, 2017).

1.6 Overview of theory underpinning the study

The development and deployment of technology is a social phenomenon (Orlikowski & Robey, 1991). As a theoretical lens underpinning, this study used duality of technology model (DoT) by Orlikowski adopted from Giddens' structuration theory to understand factors which influence quality management in the software development process.

Structuration theory has been used extensively in the development and use of technology. Structuration Theory (ST) is a theory developed by Anthony Giddens to define a unique approach to the study of social relations (Stones, 2007). Giddens emphasised on

understanding social practices across time and space and defined structure and human agency as a duality of structure and action (Giddens, 1984). Structure according to Giddens (1984) is a set of rules and resources drawn through dimensions of facilities, norms and interpretive schemes.

Structuration theory does not include technology and organisations according to several studies including Orlikowski in 1992. Orlikowski (1992) argue that technology has no influence on action rather there is duality of technology where individual or group behaviour influence technology and technology is also influenced by the behaviour hence duality of technology. The reason the DoT model by Orlikowski is most suitable for this study because software practitioners (agents) involved in product development create and recreate software products and processes. Drawing from previous studies DoT has been used to understand social phenomena about software development, software failure, quality assurance and many other topics (Jones, 2008). Duality of technology's appeal to software development lies in its focus on structure and on the processes by which structures are used and modified over time. According Poole and DeSanctis (2004) research focus has shifted to the structure of the interaction of people and technology and trying to improve processes.

Sedano, Ralph and Perraire (2017) stated that developing a software product is a complex socio-technical activity which involves people of different skill set and disciplines working together thus waste can emerge that may cause the dissatisfaction for the customer. Skills, knowledge, assumptions, and expectations about a software product and its use are drawn from training, communication, past experiences and the social environment that surrounds them (Twum-Darko, 2014). Generally, as Stark (2015) stated people, play a major role because technologies are designed, developed and managed by people (Cohen & College, 2014) and development and quality management processes are not detached from human culture. According to Murgia, Tourani, Adams and Ortu (2014) other factors such as emotions (happiness and sadness) can influence the effectiveness of proper communication and collaboration in software development.

The structuration of software quality management will lead to its institutionalisation and hence improve the delivery of a quality product.

1.7 Problem conceptualisation

Given the underpinning theory and literature review the research problem can be conceptualised as depicted in figure 1.3 below:

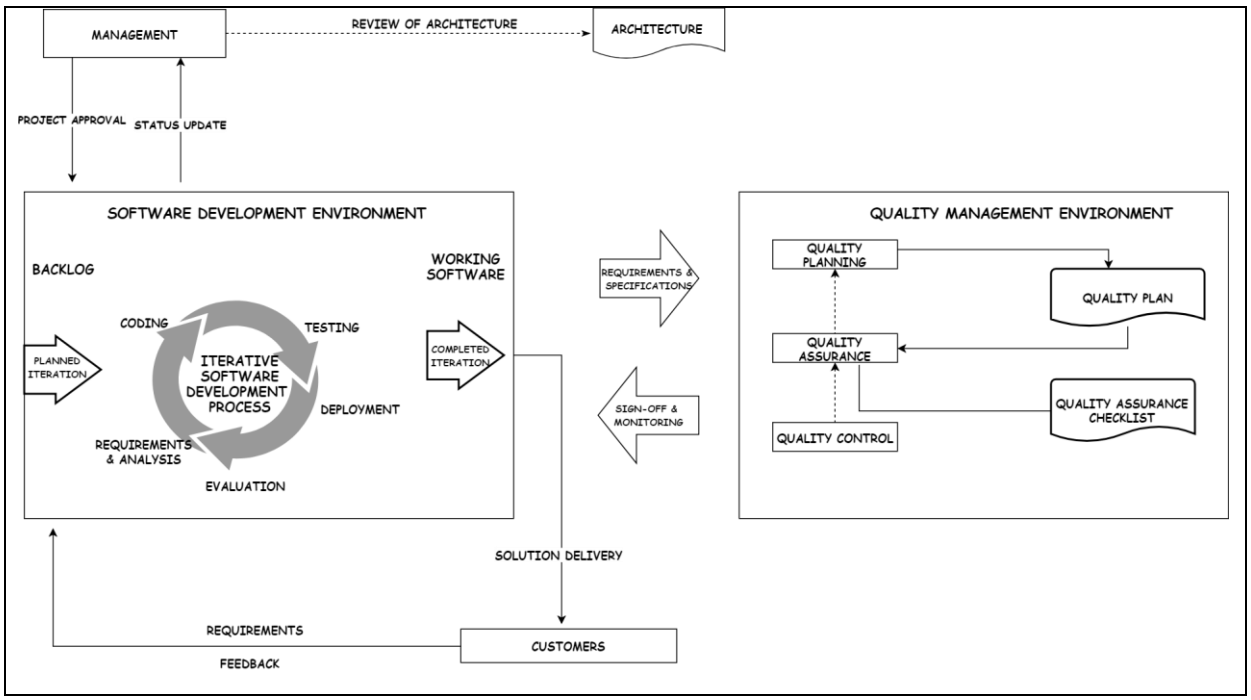


Figure 1.3: Conceptual framework

The diagram above shows the iterative process as sequential but is interleaved as new requirements and feedback come in. Some processes are repeated until an organisation has a viable product.

1.8 Overview of research design and methodology

This section covers appropriate principles of research design, methodology and collection of data used in this study.

1.8.1 Research philosophy

This study followed an interpretive philosophy stance which according to Mora, Gelman, Steenkamp and Raisinghani (2012) provides meaning to complex realities and provides in-depth insights into the context of quality management in the software development process.

1.8.2 Research design

The nature of the research and the research objectives determine the research design. It is determined by the type of knowledge the researcher intends to generate (Draper, 2004). Rowley (2002) defined a research design as a strategic framework used to conclude the research. This study used an empirical case study to get a better understanding of the research phenomena. The case study was based on a South African software development organisation

(Stitch, fictitious name). sestet is a SAAS (software as a service) organisation with a significant customer base. Effective quality management is necessary for all software product development (Sommerville, 2011).

1.8.3 Research methodology

Research methodology is a theory which determines if kinds of problems are researchable. Research methods guide the researcher on how to collect and analyse data (Johannes son & Peron's, 2014). The aim of this study was to gain deep insights drawn from the participants experiences. Therefore, an in-depth investigation was needed in a qualitative study. The rationale behind qualitative research is because it explores individual experiences (Nicholls, 2017). It draws meaning from people's words and actions.

A qualitative study also involves phenomena to do with quality whereas quantitative focuses on the quantity to derive the phenomenon (Kothari, 2004). Nicholls (2016) also argued that qualitative is more real-life focused in cases where you want to investigate an area of interest which has not fully been studied.

1.8.4 Population and sampling

The target population for this study was 24 software practitioners at stitch. Participants were primarily product managers, developers and quality assurance analysts.

The sample frame was drawn from the target population. In this study, every one of the 24 practitioners were eligible for selection for the interviews. This study used non-probability purposive sampling. According to Barreiro and Albedos (2001), the person selecting the sample makes the sample representative depending on their knowledge and roles. The purpose of this study was to get rich insight into factors that affect quality management in iterative software development process.

The sample frame consisted of 24 people. The margin of error that could be accepted was 5% with a confidence level of 95% and a 50% response distribution. The ideal sample size needed to be 21.

1.8.5 Data Collection

There are various methods for collecting data. In this study, data was collected using semi-structured qualitative interviews comprising of both open-ended and closed-ended questions guided by the dimensions of duality of technology model and quality management framework of ISO 9001. Questions were shuffled depending on how the interview was proceeding. The interviews were targeted to all the population at sestet. 22 participants responded.

1.8.6 Data analysis

Data analysis is the process of analysing data to interpret the research results and generate a conclusion (Theron, 2015). The analysis of data was done by following the trends from the patterns that emerged during the study. In this study, the interviews recorded were transcribed manually. Once transcription was complete, coding began from the transcribed information complimented by field notes recorded during the interviews. Coding is important as it helps in comparing similar patterns drawn from the data (Theron, 2015).

Themes were compiled based on the codes. Sutton and Austin (2015) referred to theming as a way codes are drawn together to better present research findings. When findings are organized for presentation each theme will represent an idea under a section of its own. Duality of technology model was then used to draw meaning from those themes.

1.9 Ethical considerations

Privacy and confidentiality: There are privacy and confidentiality concerns regarding disposal of personal data. Privacy and anonymity of individuals and the organisation participating in the research were protected.

Informed Consent: Therefore, to address these ethical concerns of both the respondents and the University, full consent was obtained from stitch and CPUT prior to the study. All participants had a choice to opt into participating in the study and were informed before the interviews took place.

Honesty: Any type of communication in relation to the research was done with honesty and transparency. Findings were reported in honesty and there was no tampering of the data.

1.10 Contribution to the body of knowledge

This study continues to build upon the previous research results to broaden and strengthen the body of knowledge by providing a general framework based on insights from ISO 9001 and Orlikowski's duality of technology model with intent to improve quality management process in iterative software product development. This research will benefit not only the selected organisation under study but all African software development organisations in improving their businesses.

1.11 Limitations and constraints

This study was a case study, used as a representation of African software organisations using iterative software development methodologies. Although the results or findings cannot be generalised across the industry, the guidelines can be adapted to apply generally across all SAAS organisations involved in iterative software development. In addition, some participants at the organisation were unable to participate in the interviews due to time constraints. Some of the interviews were rushed, and some questions were not answered because of time constraints. However, this does not affect the outcome of the study.

1.12 Delineation of the research

This research is a case study of one organisation in Cape Town, South Africa taken as a representation of SAAS organisations involved in iterative software development. Attention was given to the product development teams:

1. Product and User Experience team;
2. Development team: and
3. Quality Assurance team

1.13 Outline of the study

The rest of this research is structured into different chapters addressing different areas about this research. First background and relevant contextual information about quality management and iterative software development processes are provided in Chapter Two (Literature Review). The theoretical underpinning of the study using structuration theory is discussed in Chapter Three (Theoretical Underpinning). In Chapter Four (Research Methodology), the research design and process are outlined to address the research questions. Further on results and analysis are discussed in Chapter Five (Results and Findings) and the results were eventually summarised, and recommendations made in Chapter 6 (Conclusion and Recommendations).

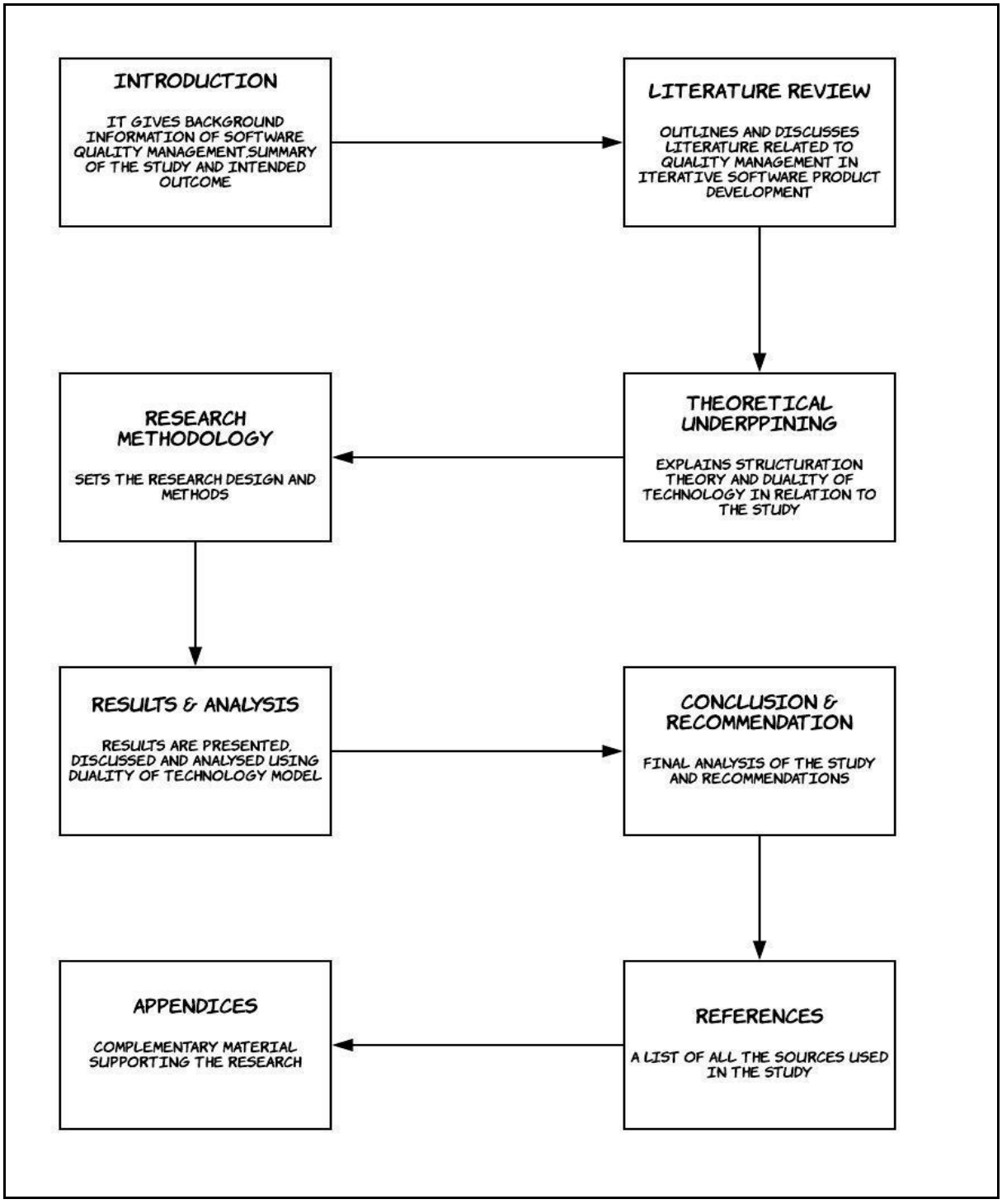


Figure 1.4: Dissertation structure

1.14 Summary

This chapter outlined the structure of the research study conducted and research context. The aim of the study was to introduce the importance of quality software product delivery and its benefits to both the customer and the organisation developing the product. This chapter explains how rapid technology is changing and how complex software products are becoming.

With this development, the pressure is put on organisations to innovate while making sure that they have effective software quality management systems.

Therefore, the aim of this study was to identify some of the quality management challenges that are faced by product development teams of the selected organisation to determine what factors influence the quality management process of software product development. The organisation under study uses an iterative process to deliver its products. The intent of this study was to produce a guideline to mitigate some of these challenges faced by the selected organisation as well as other similar organisations in the industry.

The next chapter will outline literature produced from related studies, with regards to software product development processes, quality management and socio-technical processes embedded in software product development.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

The previous chapter was an introduction of the study giving an outline of the research problem of inadequate quality management in iterative software development process, the research objective and the intended outcome. In this era, software development environment is faced with a rapid change in customer requirements, complexity and innovation. In the consumer world, customers are becoming more product navigators with the power of choice. To cope with the rapid change in the business environment, some studies have proposed standardisation of software quality management. However, no extensive evidence is available to support this proposal.

Most studies done concur on the idea that the quality of the product depends on the quality of the software development process (Mansoor, Bhutto, Bhatti, aamir Patoli & Ahmed, 2017). According to Kassab, DeFranco and Laplante (2017), quality is a very elusive concept which can be approached via different means depending on one's understanding. Evidence that relates to software processes to quality taken from different samples is usually contradictory. There is little data regarding actual perception from software personnel towards the quality requirement of software products.

Sowunmi, Misra, Fernandez-Sanz, Crawford and Soto (2016) point out that one of the factors that affect the establishment of a quality process is the absence of a framework for quality management in an organisation. Thus, adequate quality management becomes a necessity as it promotes competitiveness for the organisation (Femmer, Fernández, Wagner & Eder, 2017; Highsmith *et al*., 2001).

In this chapter, the common challenges of software development are outlined followed by a discussion on iterative and incremental models to software development. Next, quality management is discussed. The last section of this chapter outlines the related studies.

2.2 Software product development overview and common challenges

Software products are touching many aspects of our daily lives defining the behaviour of things that surround us (Bertolotti & Hu, 2016). The development of software products has brought us to a point where we cannot live without some of them. It is important that we control how

they are developed for them to be of the right quality (Ebert, Abrahamsson & Oza, 2012; Stark, 2015). In this digital era, software products are being developed more than before and many more will be developed in the future via different platforms including mobile applications and websites (Tan *et al.*, 2009). Software organisations have to stay relevant and constantly need to improve their software delivery processes.

There are certain continuous practices that Fitzgerald & Stol (2015) described as necessary for software organisations to sustain the rapid changes in technology. These include continuous planning, integration, verification, testing and quality management, compliance, security, use, trust, execution time, monitoring and improvement in both the product and the development process.

Lee and Chin (2017) asserted the need for software delivery processes to be systematically designed, developed, implemented, managed and improved. Also making sure the product development teams are trained and follow practices that conform to business needs and regulations. This is to ensure that high quality software products are produced. However, software development teams do not want to focus too much on process as it is perceived to reduce productivity (Clutterbuck *et al*., 2008).

Schüler, Trogus, Feilkas and Kinnen (2015) also stated one challenge that organisations constantly face is that of software decay which demands high maintenance. If there is high developer turnover in the organisation, it is imperative therefore to have properly documented software delivery processes in place that makes it easier for new software developers to come in and take over a project. Another challenge which (Ramakrishnan & Manjula, 2016) put forward is the growth of the product. As the product grows, small defects tend to be ignored. These defects if ignored reduce reliability and trust of the product.

Sommerville (2011) assert that ISD processes usually follow an informal approach to change and quality management. Mateen *et al.* (2017) state that management often does not take the process seriously. According to Jamsutkar *et al.* (2012) having an effective quality management process ensures business objectives such as faster time to market, competitive advantage and customer satisfaction are met. In the long term, it reduces costs because any defects are located and fixed sooner (Sowunmi *et al.*, 2016).

2.3 Software product development processes

The development of a new product begins with requirements analysis through collated needs of customers, analysed, and evaluated. Blueprints for the product are developed at the design stage based on the customers' needs and the product specifications (Alhassan *et al*., 2017). Iterative and incremental approaches are becoming popular due to their flexibility regarding implementing changes and traditional approaches such as waterfall are becoming obsolete.

2.3.1 Waterfall approach

The traditional 'waterfall' methodology is a sequential development approach which flows from top to bottom. The development phases of the waterfall approach as depicted in figure 2.1 are systems requirements, software requirements, architectural design, coding, testing and maintenance.

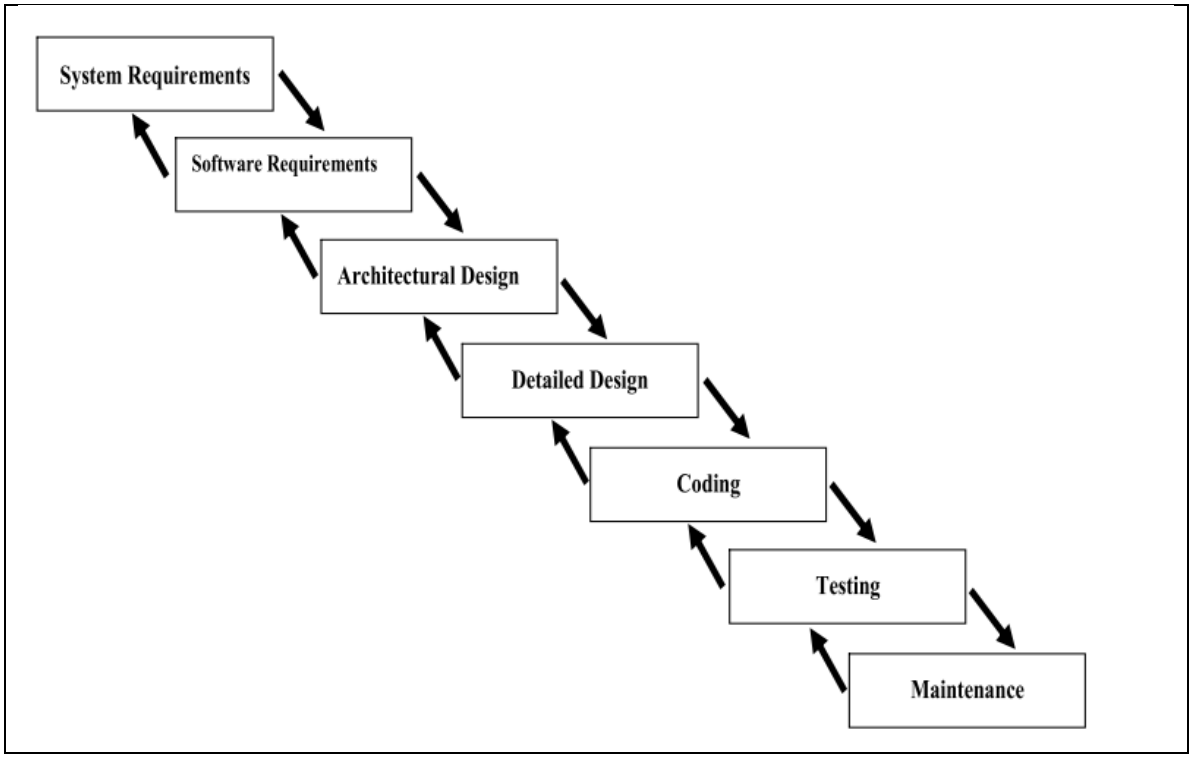


Figure 2.1: Waterfall approach

(Adapted from Mohammed *et al.*, 2010)

Mohammed *et al*. (2010) opine that the waterfall approach reinforces good habits of design before code. In general, it has a potential to provide a straightforward, systematic, and organised process in the software product development suitable for long-lasting complex software products where requirements rarely change (Flora & Chande, 2014). However, Moniruzzaman and Hossain (2013), in their study argued that the business world is now rapidly changing, requirements will keep on changing such that organisations are forced to do

continuous delivery of the software products. Mohammed *et al*. (2010) further argue that software product teams should not make assumptions that the requirements will not change because they do. If organisations are going to release software at the end of the process, they are bound to encounter serious errors which might make the software product unusable. Fitzgerald and Stol (2015) also argue stating that the waterfall approach is risky in that quality management practices are towards the end of the project. There is too much administration and excessive documentation involved which could be costly and stand in the way of working software, or the product could have serious defects (Hamman, 2009; Mohammed *et al*., 2010; Karout & Awasthi, 2017).

Abrahamsson *et al*. (2007) argued that traditional approaches are not practical. Customer input is most often received late because it is required in the last stages of development and this causes the development cycle to drag on as the input is implemented. These disadvantages are claimed to be advantages for iterative and incremental approaches by several studies such as (Kruchten, 2008; Silva & Kazmierkowski, 2016; Karambir & Sharma, 2016; West & Grant, 2010). They argue that an iterative approach brings higher quality products in shorter periods of time that satisfy the needs of customers.

2.3.2 Iterative approach

The major reason why ISD approach is replacing traditional plan-based software development methods is that it helps improve the software development process to meet the rapid changing business environments (Moniruzzaman & Hossain, 2013). An iterative approach follows a more effective framework than traditional approaches which makes a high-quality software product that meets customers' needs (Abrahamson *et al*., 2007; Jinzenji, Hoshino, Williams & Takahashi, 2013).

Iterative approach is referred to by some authors as an Agile approach. In 2001 a group of software practitioners gathered and came up with four main values for what is popularly known as the Agile Manifesto. The values according to Fowler and Highsmith (2001) are:

(a) Individuals and interactions over processes and tools
(b) Working software over comprehensive documentation
(c) Customer collaboration over contract negotiation
(d) Responding to change over following a plan

Agile is suitable for when complete requirements of the product are not known in advance or are expected to change or when a minimal viable product (MVP) is essential to meet the

customer needs (Doshi and Patil, 2016). Jalote and Agrawal (2005) stated that feedback from each complete iteration and changing customers' needs helps improve future iteration and the software delivery process in general. Highsmith *et al.*, (2001) state that with ISD, the cost of change is reduced throughout a project.

Some of the examples of ISD processes include Extreme Programming (XP), Scrum, the crystal process family, adaptive software development (Highsmith *et al.*, 2001) and the agile unified process (AUP) (Turk, France & Rempe, 2005). Knippers (2011) point out that the most common methodologies which complement each other are XP (code aspects) and Scrum (project management). However, Flora and Chande (2014) contend that choosing the right methodology or methodologies is always a challenge. XP and Scrum methods help increase productivity, product quality and reduce development cycle times and time-to-market (Dingsøyr & Lassenius, 2016). In software product development, continuous improvement is proving to be the key to success in the business world. Iterative development methodologies have within the development, a step which allows the gaining of information to improve the development process and quality of the product (Oberscheven, 2013). Iterative and incremental development methodologies call for product teams to produce the first delivery in weeks to achieve an early win and get constant feedback. When releases are small, changes are less costly when change is required unlike when releases are larger. Quality is constantly improved through learnings from previous iterations (Dingsøyr & Lassenius, 2016).

Silva, Soares, Peres, de Azevedo, Vasconcelos, Kamei and de Lemos Meira (2015) argued that although ISD approaches focus on working code, they should not ignore processes and documentation. The important things include collaboration in product development, responding quickly to changes and ideally customer collaboration. However, Turk, France and Rumpe (2014) argued that some product teams find it difficult to adopt iterative product development mainly because it is difficult for organisations that develop long-lasting complex software products to be agile. The degree of agility an organisation depends on several factors which include experiences gained from previous projects, change in requirements and development environments. However, no process is purely agile because there is usually a process framework guideline if product development teams need to follow. Shrivastava *et al.* (2010) stated that ISD provides agility to an organisation to respond to changing market needs, works well with distributed teams if knowledge sharing is done effectively. Some of the advantages mentioned by Knippers (2011) of ISD methods are lower defect density, increased team communication and morale. The focus on collaboration helps bring value to the user in the most efficient way (Popovic, 2015). However, Jinzenji *et al.* (2013) have argued that ISD methodologies do not provide a way to manage and measure quality making it difficult for other

organisations in critical domains to adopt this new approach to software product development. Abrahamson *et al*. (2007) further argue that frequent release may bring confusion rather than clarity. The use of an ISD process is still risky, complex and unique and determining the necessary process that will produce a quality product is not easy (Borque & Fairley, 2014). The process requires experience and skill to be effective (Mohammed *et al*., 2010) and it needs to be adjusted to suit a situation to meet the quality required (Huo *et al*., 2004). Shmueli and Ronen (2017) have criticised ISD as to be weak in the area of requirement engineering stating that welcoming late changes may cause excessive software engineering.

2.3.2.1 Scrum

Oberscheven (2013) described a scrum team comprising of a product owner, a scrum master and software developers. Product development is in small cycles (iterations), usually for 2-4 weeks sprints with select features from the product backlog ready to be deployed. Any new features would have gone through all the SDLC stages (Hamman, 2009). Within each sprint cycle, there are specified backlog features that need to be delivered. Every day the product development team meet for about 15 minutes with a product owner for collaboration and integration (O'Leary & Gordon, 2009). At the end of the sprint, a shippable increment is produced and presented to the product owner (Popovic, 2015). The product owner makes the final decision to say if the work item is ready for shipment according to requirements (Oberscheven,2013). The team will have a retrospective meeting to analyse good, bad and what they need to improve in the development process (Popovic, 2015). This process is summarised in figure 2.2 below.

Figure 2.2: Scrum approach

(Adapted from Mendez & Pinto-Roa, 2018)

Turk *et al*. (2005) assert testing and customer feedback as the most important quality control mechanisms in ISD approaches. In Scrum, review meetings help to ensure that a quality

product is produced. Scrum is designed to help product development teams improve their software delivery process. It is faster, better, cheaper and more responsive to change (Hamman, 2009). Scrum is the best approach to project management at the same time increasing the success of software development (Silva *et al.*, 2015). Scrum helps complete complex projects (Popovic, 2015).

As much as there are so many pros to the scrum methodology, it fails on the issue of criticality (Flora & Chande, 2014) and it does not offer guidance on meeting the quality requirement, a team will have to adopt another approach for it to be effective (Khalane & Tanner, 2013).

2.3.2.2 Extreme Programming

Extreme programming (XP) is one of the common ISD methodologies that have been associated with aiding quality software delivery. Its practices and life-cycle are summarised in figure 2.3 below:

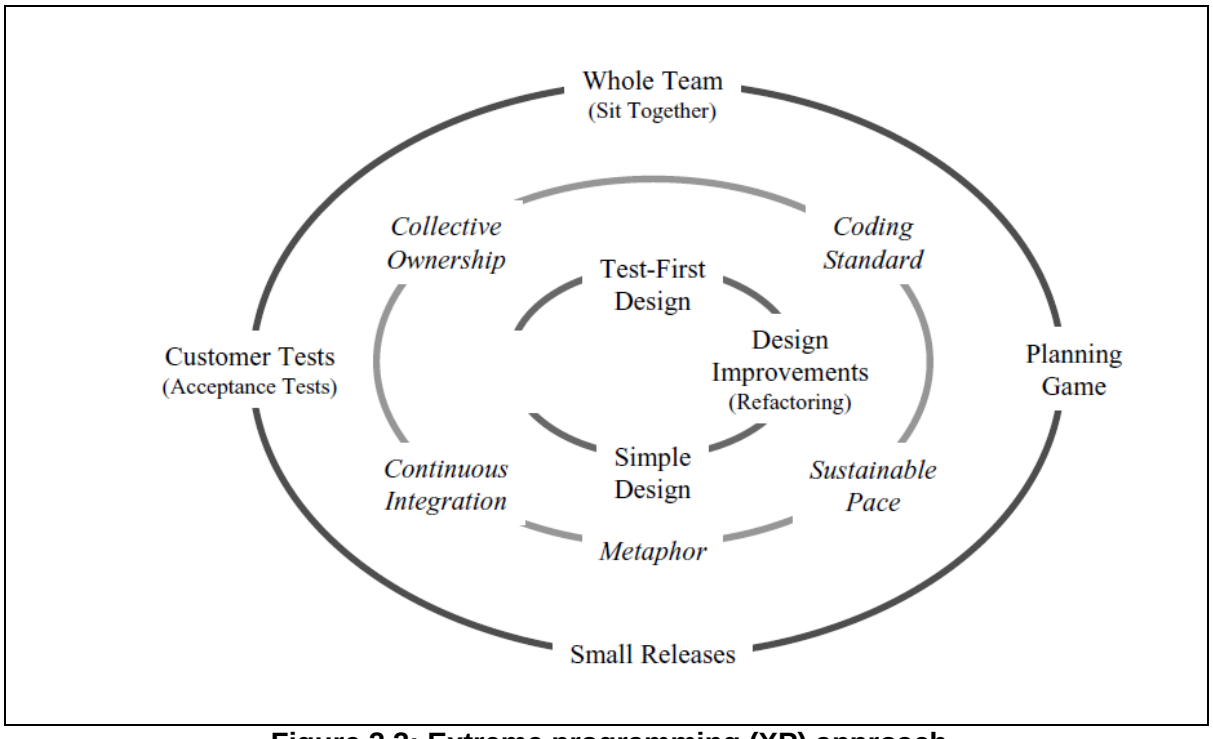


Figure 2.3: Extreme programming (XP) approach

(Adapted from Lindstrom & Jeffries, 2004)

Mohammed *et al*. (2010) listed XP practices as incremental planning, simple design, small releases, test-first development, refactoring, pair programming, collective ownership, continuous integration, sustainable pace and onsite customers. However, they also mentioned that some of these practices such as pair programming are costly, the process requires skill

and is difficult to scale up to large projects. Pair programming and informal reviews do not guarantee that the product is of quality. Hence the need for combining these techniques with a more formal approach (Abrahamson *et al*., 2007). Turk *et al*. (2004) also argue that it relies on source code and product knowledge lies in developers' heads who developed the product such that when they leave the organisation, this could be costly. Further on, Flora and Chande (2014) argue that XP is not suitable for big and complex projects, requires developers to collaborate and testing is done by one person.

Another aspect is that of continuous integration, Fitzgerald and Stol (2015) state that it requires a link between development and operations (DevOps). Claps, Svensson and Aurum (2015), stated that in agile, DevOps had widened its effectiveness by also bringing IT operations together with developers, testers and product people thereby improving the quality of delivery of the software product.

2.5 Software quality management

Silva *et al*. (2015) state the need for ISD approach to adopt additional practices; it alone will not be able to meet the level of quality required. Introducing quality management process in software development requires following a universally acceptable standard or model and defining what quality means in an organisation (Yao & Lee, 2004). Software quality means meeting the quality requirement defined by the customer.

As defined by Ebert *et al*. (2012), quality is the *"the ability of a set of inherent characteristics of a product, service, product component, or process to fulfil requirements of customers.*" Quality management is managing the software product quality and its development process (Pandey, Saxena & Pandey, 2013; Colomo-Palacio, Soto-Acosta, Mishra & García-Crespo, 2011). Quality management (QM) defines quality goals for the software products, creating plans, putting measures how the goals will be achieved, monitoring and adjusting as fit to satisfy customers (Aregbesola *et al.*, 2011). Sommerville (2011) describe QM purpose as to ensure that software has a low number of defects. Pandey *et al.* (2013) assert that QM bridges the gap between software practitioners including customers, architects, software developers and testers. However, Sowunmi and Misra (2015) argued that even if product teams are aware of software quality practices, they do not adhere to the quality standards nor have people that ensure that the standards are adhered to.

In some organisations, particularly large organisations there is a separate independent department responsible for quality management and it is commonly referred to as the quality assurance (QA) team. In smaller organisations, it is more informal, and the development team

usually also performs this role (Sommerville, 2011). However, Sowunmi *et al.* (2016) state that it is the responsibility of every team member to ensure that the product meets the customer expectation. Aregbesola *et al*. (2011) argue that QM process should be separated from software development. However, it is evident from the literature review that the two should complement each other.

Dovleac and Ionică (2017) explored quality management techniques embedded in ISD. The authors stated the following quality attributes in ISD; focused on constant improvement of delivered, working software is more important than documentation, progress is monitored through daily meetings and results, works with user stories, proactive response to change, sustainable development, simplicity is essential, allows teams to self-organize and gives freedom to team members in choosing architecture, requirements and design and gives power to team members. Figure 2.4 below shows the new proposed agile approach inspired by the plan-do act.

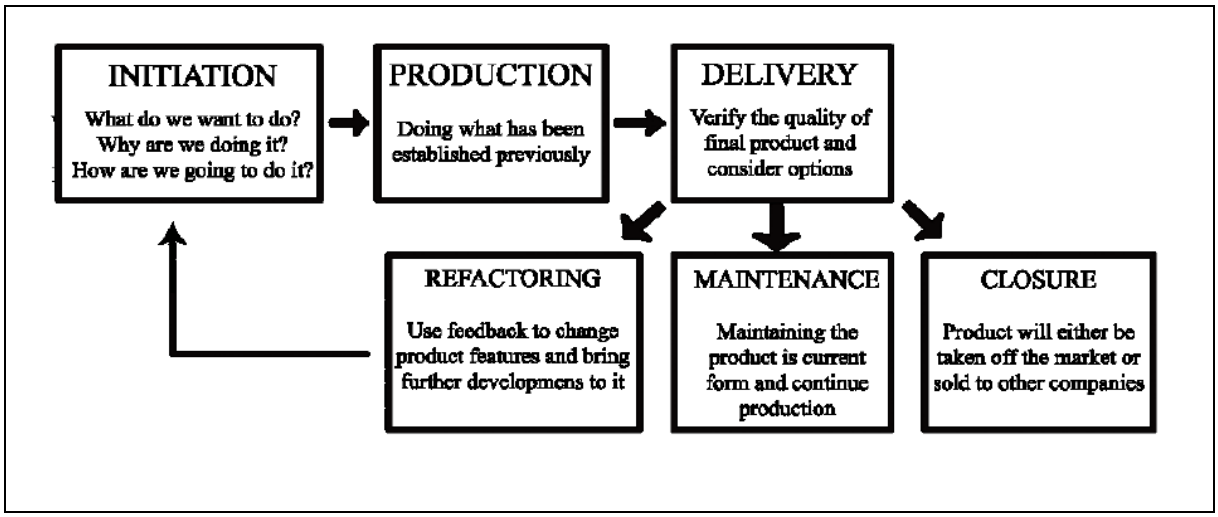


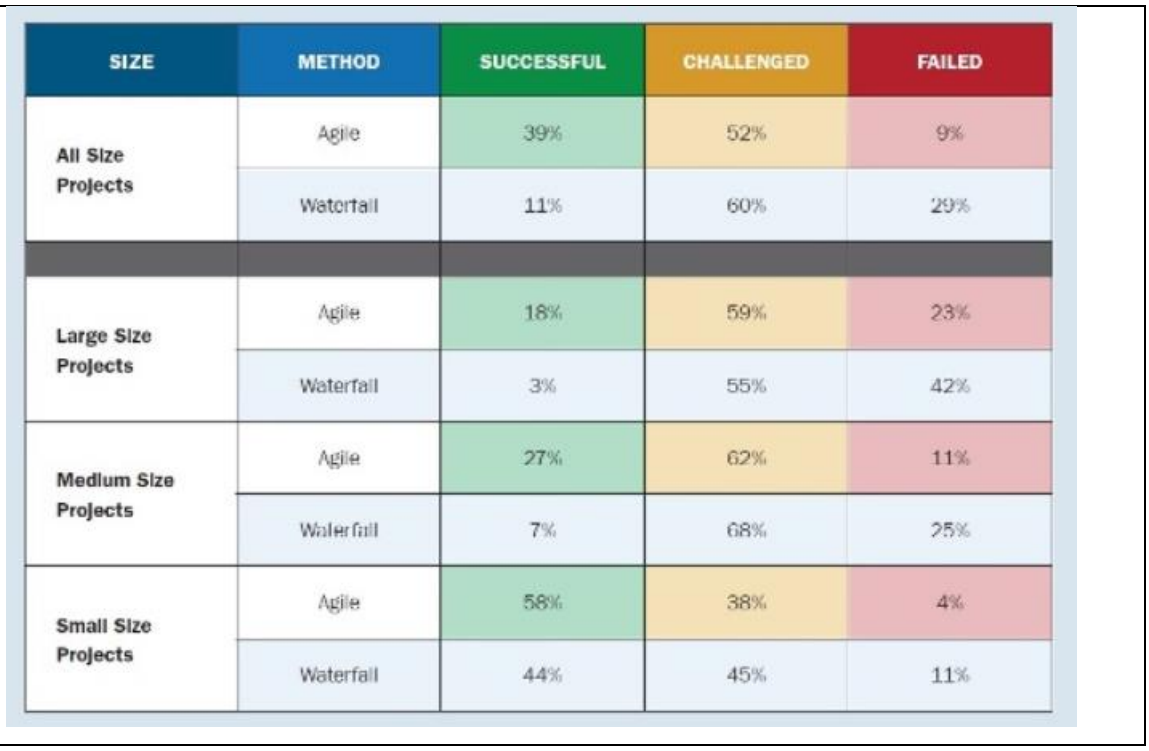
Figure 2.4: Agile life-cycle with quality management

(Adapted from Dovleac & Ionică, 2017)

Alhassan *et al*., (2017) state that the effectiveness of a QM process is determined by how performance and results are measured. The quality management system must be designed in such a way that convinces developers to adopt quality management practices (Janes, Lenarduzzi, & Stan, 2017). Femmer *et al*., (2017) argue that the QM process framework can help organisations improve performance and satisfy employees, suppliers and the customers.

However, there are also certain challenges that are incurred. Hossain (2018) identified inadequate software quality management as the major challenge firms face today which reduces the firm's growth. The issues can be put into the three categories general, stakeholder

perspective and requirements. Figure 2.5 below extracted from a report published in 2015 for the period between 2011 and 2015. The report shows software development project statistics for agile vs waterfall approaches. It is evident that quality is still a major issue in ISD especially for large projects.



SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

Figure 2.5: Success, challenged and failure of software projects

(Adapted Chaos Report, 2015)

According to a study done in Australia by Alshammri (2013) on Huntington Bancshares, Inc. and Jordan Telecom. It was stated that software developers do not pay attention to quality of the software products and this has caused problems for these two organisations. The author suggested that they should be people assigned to the process of quality management practices for each iteration to reduce software failures.

Another challenge that Giardino *et al*. (2016) stated was that of technical debt caused mainly by releasing software quickly without proper quality management processes. The author further stated that technical debt is costly for organisations in the long term.

Zope *et al*. (2016) also argued that practices provide guidance but do not include all the stages of the SDLC so that people playing different roles in the process are aware of their contribution to software quality and assuring it through related activities and design. Zimon and Malindzak (2017) argue that this is time-consuming and costly. Although the steps to an effective quality

management process may be known, Mateen (2017) argue that most organisations do not properly follow the best practices and are not properly owned. Abrahamsson *et al.* (2017) also stated that developers sometimes lose track of quality fluctuations within the software product being developed.

2.5.1 Quality management components

The four components of quality management are planning, control, assurance and improvement (Nanda, 2016).

2.5.1.1 Quality planning

Quality planning involves the creation of a quality plan for a project (Sowunmi *et al*., 2016). Sommerville (2011) further state this includes selecting applicable procedures and standards for a project and modifying these as required. Sowunmi *et al*. (2016) opine that it involves defining a quality assessment process to help evaluate the process at the end and involves defining the customer's expectations. If the quality is not defined, developers might develop something different.

The quality plan needs to define what the product is and the quality expectations of the product with each iteration. Adding on, release dates need to be also stated and communicated. Quality plans may differ with the size of the project. Some do not need a big formalised process. Thus, documentation should not be heavy for small projects otherwise people will not read it (Sommerville, 2011).

2.5.1.2 Quality assurance

Quality assurance (QA) is a repeatable process which needs to be part of the development process to ensure that software standards and procedures are followed. Quality assurance is "a set of activities designed to evaluate the process by which software work products are developed and or maintained" (Aregbesola *et al*., 2011). The success of quality assurance is dependent on the quality plan (figure 1.2). Here metrics are set which quality assurance will validate against to make sure the software product does what it is intended to do (Lee, Ko, Park & Kim, 2014).

It goes beyond testing to control and ensure that the product meets the customer expectations (Sowunmi *et al*., 2016). Aregbesola *et al.* (2011) contend testing alone is not enough to meet business expectations, quality management needs to be the focal point. Testing only validates

if the product meet customer expectation for functionality and performance before deployment. Quality needs to be managed from the beginning not just at the end so that any issues are identified early and are rectified (Stark, 2015). Quality assurance ensures that the software is functional and well documented for easy maintenance.

Organisations need to follow standards to ensure that during the delivery process defects are removed before a new product is released (Sowunmi *et al*., 2016). Code reviews add additional safety to automated testing. McIntosh, Kamei, Adams and Hassan (2016) stated that the software code review involves the practice of having other developers review the code before release to ensure quality requirements are met. Many teams have integrated code review tools to version control systems such that every committed change may be reviewed by more people (Mäkinen, Leppänen, Kilamo, Mattila, Laukkanen, Pagels & Männistö, 2016). Baum, Liskin, Niklas and Schneider (2016), also state that code review is an efficient quality assurance practice. However, not all organisations have adopted this technique in their development processes because of strict deadlines and resource availability.

Patwardhan (2016) carried out a study in which he proposed a structured unit testable templated code for effective code review process. The author argues that if a developer is not familiar with scripting languages and is used to object-oriented scripting languages they may make a lot of mistakes which may result in more code review iterations and inability to identify system wide implication of the code. If senior developers are not engaged in code reviews this may also result in having poorly written code. Developers often engage more on requirements than actual code. Code discussions are often around syntax and formatting issues.

Kaur and Sengupta (2011) state that the purpose of testing is to find defects and to have them removed before the product is released to the customer. Although developers test the product during development, user acceptance testing helps ensure that the product meets the business requirements. Tuteja and Dubey (2012) state that testing is an important aspect to ensure quality. However, Alshammri (2013) argue that some organisations do not pay much attention towards quality assurance issues and thus eventually end up losing money and time. According to Fitzpatrick *et al*. (2004) quality assurance alone does not fully address the need for quality management throughout the product life-cycle.

2.5.1.3 Quality control

Quality control is the process of monitoring the software development process with the intent to ensure the quality plan is followed and standards are met by the product team. Ideally, this is to ensure that quality goals are met (Pressman, 2010). In practice, a quality control process

will only be effective if the quality plan is accepted by everyone involved (Turk *et al*., 2004). Although ISD has inbuilt quality management processes, Baig, Shah and Sajjad (2017) suggested that quality in ISD can be improved by adding quality control techniques depicted in figure 2.6 below:

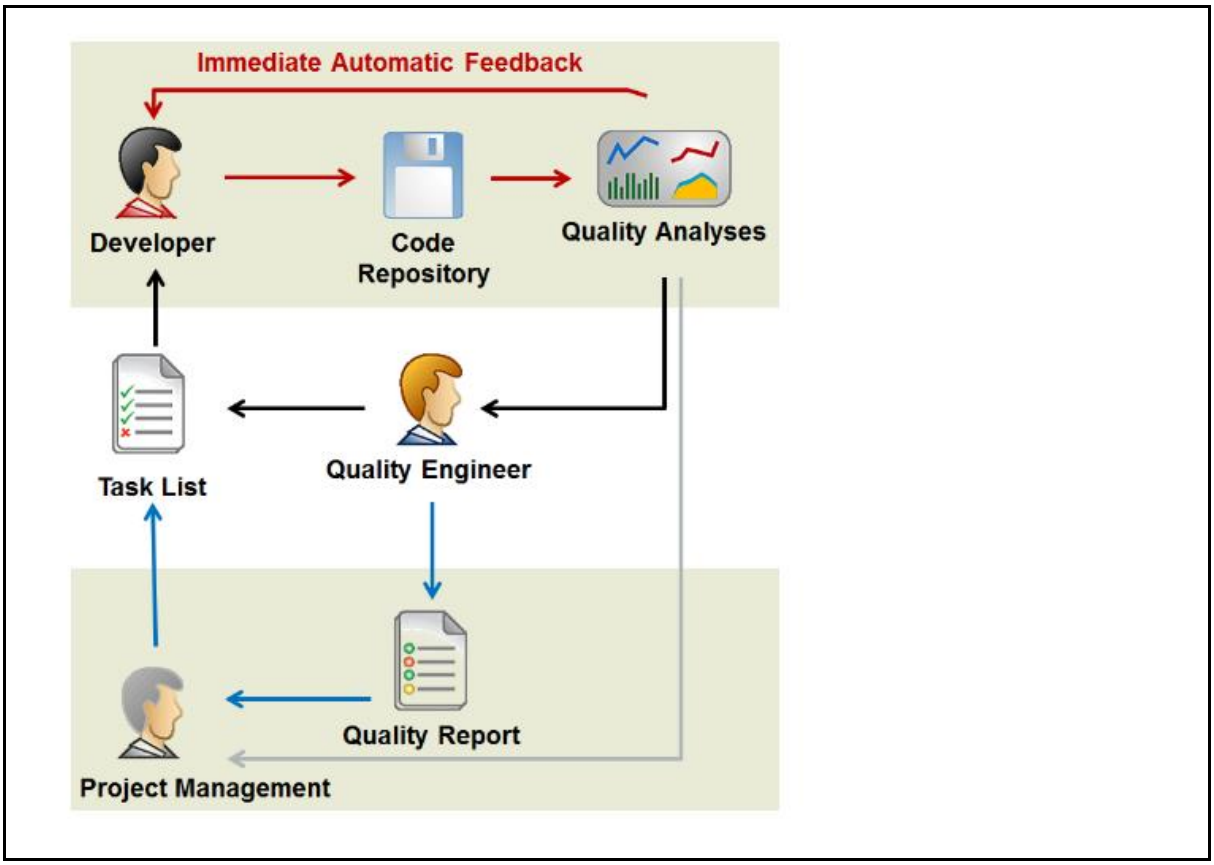


Figure 2.6: Quality control process

(Adapted from Torgus & Kinnen, 2015)

Figure 2.6 shows a continuous quality control process so that quality is measured throughout the software development process not right at the end towards release.

2.5.2 Software quality improvement approaches

According to Jovanovic and Shoemaker (1997), successful software quality management includes the whole development process not just control of the final product. An acceptable model of reference is needed when creating a software development system.

There are four quality standards for software product delivery (Rizvi, 2014):

(a) CMM (Capability Maturity Model) has five levels of maturity that determine the effectiveness of the organisation's software delivery process. Level 1 is chaotic with not much process in place. Level 2 has a bit of structure and processes which can be repeated. Level 3 has standards and training in place, there is understanding. Level 4 there is metrics tracking and productivity and in level 5 there is continuous improvement and better management of new processes and technologies.

(b) IEEE (Institute of Electrical and Electronics Engineers) is a standard for quality assurance plans.

(c) ANSI (American National Standard Institute) is an American standard for promoting standards and conformity assessment systems and protecting their integrity.

(d) ISO (International Organisation for Standards) is a series of standards. ISO 9000 series provides the guidelines for software quality management. ISO 9001 gives the structure for software organisations to develop quality management systems (Tomar & Thakare, 2011). This study focuses on ISO 9001 which is outlined in the next section.

2.5.2.1 ISO 9001

IS0 9000 series first appeared in 1987 (Coallier,1994). Sommerville (2011) defined ISO 9001 as an international set of standards used by software development organisations. Qasaimeh and Abran (2013) state that ISO 9001 requirements originated from manufacturing sector but, have also been adopted in the software development sector. The authors argue that the requirements are independent of the development methods used. The ISO 9001 standard is a quality management system designed to help organisations meet the needs of their customers and other stakeholders (Popovic, 2015). Figure 2.7 below illustrates ISO 9001 process approach.

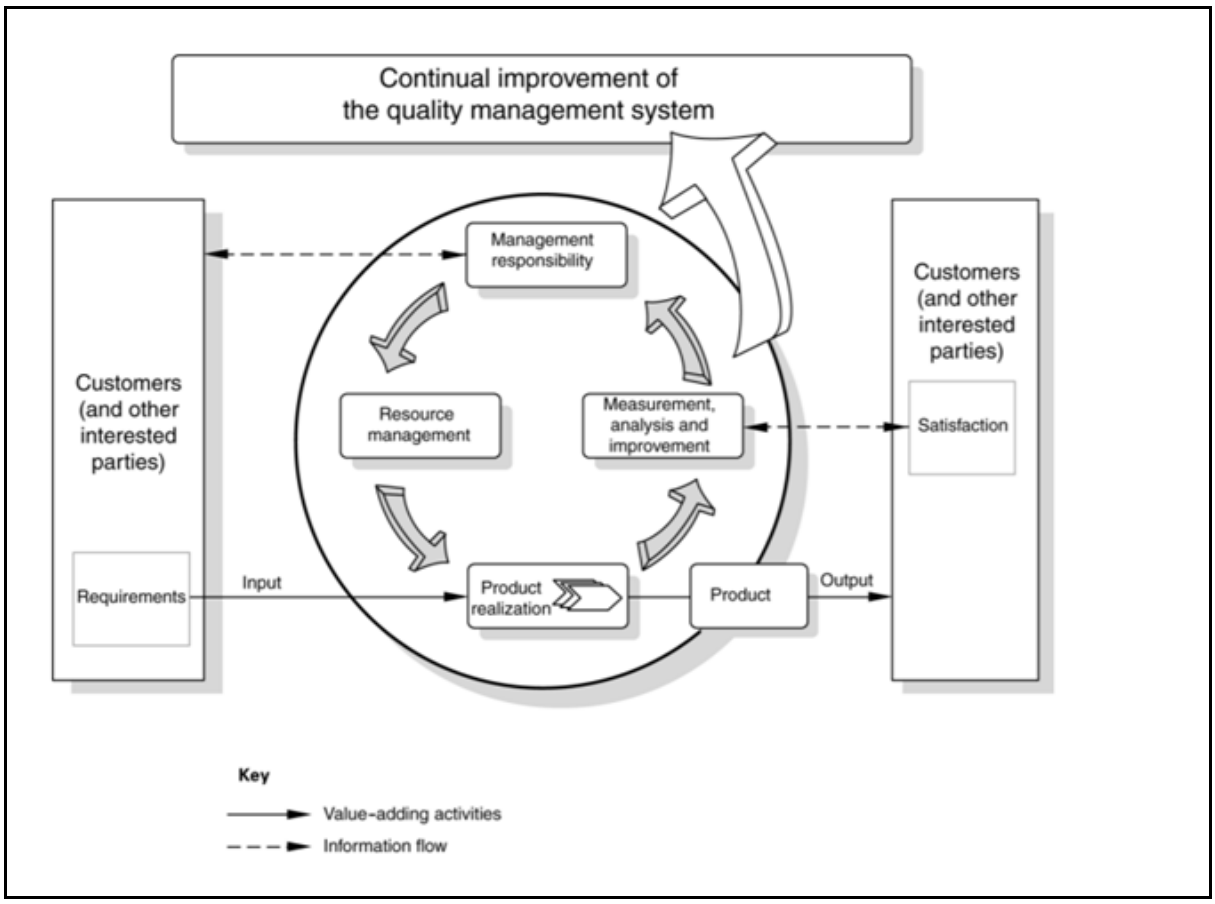


Figure 2.7: ISO 9001: 2015 Process Approach

(Adapted from ISO, 2015)

According to Rizvi (2014), ISO 9000 describes quality management in general terms. The key elements needed for implementing effective quality management are planning, assurance, control and improvements determined by the structure of the organisation, procedures, processes, and resources.

It involves defining quality standards, developing a quality culture and ensuring that the product produced is of quality (Sommerville, 2004). Stalhane and Hannsen (2008) summarise ISO 9001 requirements: First, an organisation must have a quality assurance management process. Next, it is the responsibility of management to ensure that the product is of quality. Thirdly, that the organisation should have documented process or processes. Lastly, that it must produce a process that can be reviewed for acceptance and used as proof of conformance. Non-conformances of process or product must be reported, analysed and lead to an improvement of the software delivery process.

A lot of studies have criticised the standard because it does not provide how the quality management should be implemented but some believe it's a good start when an organisation

wants to start or improve their process. Popovic (2015) opine ISO 9001 is designed to help meet customers' needs. Implementing ISO 9001 certifications could be beneficial for a small and medium organisation as it may assist them in sustaining and improving the quality of the development process and software products and in attracting new customers if they know that the organisation is ISO 9001 certified (Stalhane and Hanssen,2008).

ISO 9001 gives structure to help ensure ISD processes are followed. However, it does not equate to quality. Mcmichael & Lombardi (2007) identifies the major practices that need to be documented with ISO 9001 requirements as; software testing, configuration management, defect tracking, internationalization, product maintenance, requirements management, and release management.

ISO can successfully be merged with ISD models. Popovic (2015) explains the idea of applying ISO 9001 to ISD processes focusing on XP. The study argued that aligning Scrum and XP brings better alignment to ISO 9001 but is not necessary to do so. ISO 9001 implementation may help in adherence to ISD practices.

Although ISO 9001 has gained a good reputation over recent years, there are questions which still linger whether it works well with agile organisations. Galetto, Franceschini and Mastrogiacomo (2017) argue that ISO 9001 may not have effectiveness on the financial performance of an organisation and certification does not recognise an organisation that apply the principle of ISO 9001. Although many new studies are trying to reinforce the relevance of ISO 9001 today, back in the 1990s, the model was criticised. Some of the shortfalls of this requirement included its failure to suit the complex development of software product with ever-changing requirements. ISO 9001 does not reinforce the need for strong customer orientation, weakly addresses how an organisation know the right product for the user. (Coallier, 1994).

In figure 2.8 below is the conceptualisation of a quality management process in iterative software development. Customers provide requirements and feedback to the software development organisation which is fed into the product backlog. Prioritisation happens, and tasks are scheduled into iterations. Management approval is required and review of the current architecture to make sure the new changes can be supported. The software development process entails requirements analysis, coding, testing, deployment and evaluation.

Quality management acts as an independent check to make sure the ISD process is properly followed. Quality management is part of the software development process in that a quality plan is created based on requirements and feedback from customers and model of practice to make sure that correct product is developed to the customer. Throughout the software

development process control checks are established to ensure the right product is delivered to the customer.

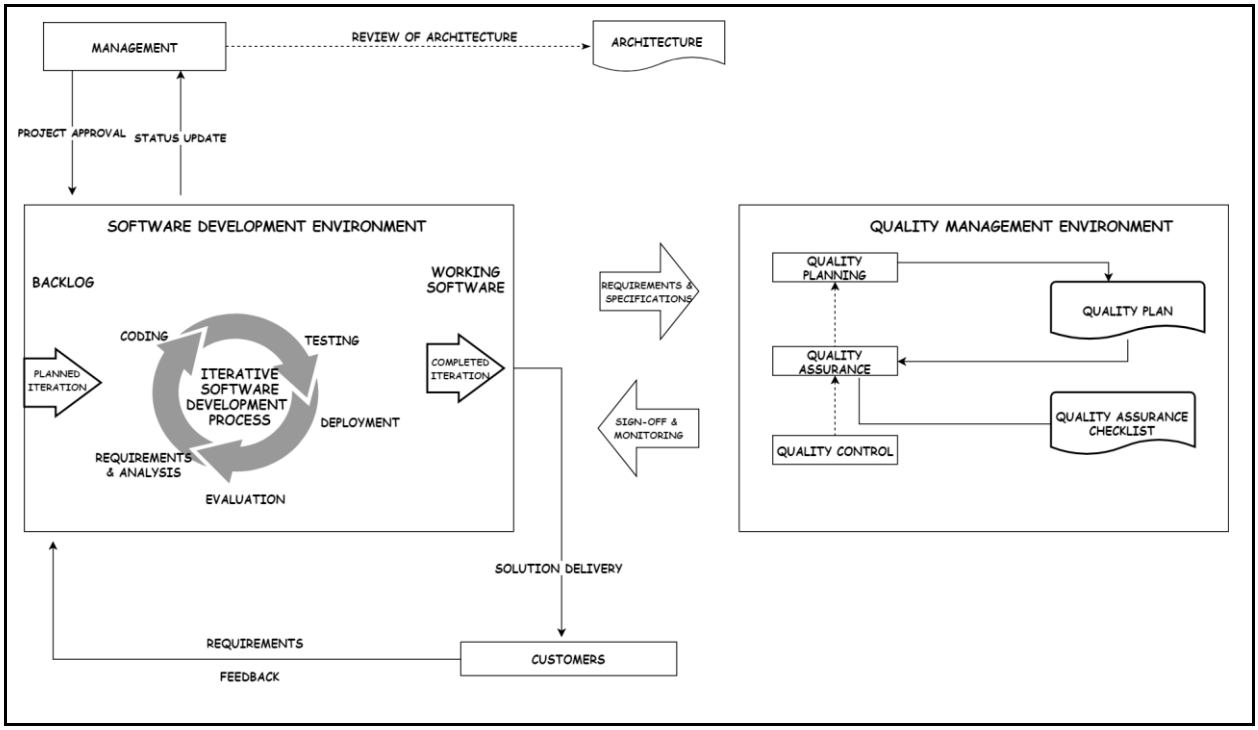


Figure 2.8: Conceptualisation: Quality management in iterative software development

2.6 Related studies

McFall, Wilkie, McCaffery, Lester and Sterrit (2017) carried out a study to discuss the current software development and process improvement practices in Northern Ireland software industry. The problem which the authors found was the lack of detailed state of practice. The outcome of their study was to understand the current state and to suggest the need for staff to get formal qualifications in software process improvement models and methods which would permit meaningful engagement in software improvement activities within the local industry.

Claps *et al.*, (2015) did a study on the Australian based high-tech organisation, Atlassian and found that the product development team adopted a combination of software development practices which are Scrum, XP and Kanban techniques. As a result, the organisation was able to deliver quality software daily though continuous delivery.

Kassab *et al*. (2017) did a comprehensive survey of software professionals to discover software process to quality practices. The study focused on the influence of quality requirements on the software architectural decisions in practice. In a competitive industry with the customers having so many product alternatives, software quality requirements become

more important in distinguishing between the competing products. In practice, quality requirements receive little attention. Software systems get redesigned because they are slow or unusable. In general, quality requirements influence architectural decisions more than functionality.

Oberscheven (2013) designed a model which measure software quality of the development process. The quality process is related to the reliability and predictability of the development process. However, a holistic approach was provided which is not suitable for one organisation with specific needs. Camison and Puig-Denia (2015) did an empirical investigation in their study using the structural equation modelling technique and six models from a sample of 550 Spanish organisations. The findings indicated that the implementation level of quality management has no influence on an organisation's innovation it only helps improve an organisation's processes.

According to Zhang, Ye and Lin (2014) quality management has widely been promoted. However, its implementation has led to failures. As suggested by many studies to put in place QM practices to suit organisations needs to avoid failure and improve performance this has not been fully investigated. Zhang *et.al.,* (2014) empirically discovered that the benefits of different QM orientations depend on the level of competition and the rate of product change.

Scarpino and Kovacs (2008) did a case study on quality management and found out that the organisation was more into software testing than following a proper quality management process. Ghobadi (2015) investigated the importance of collaboration and knowledge sharing in software development since software products continuously emerge from intensive and iterative development and quality assurance cycles that involve many people who may have opposing priorities. When there are challenges for people with different ideas it makes it difficult for the organisation to adopt proper processes.

Sowunmi *et al*. (2016) did a qualitative study on quality assurance for Nigeria in comparison with Turkey. The outcome of the study was that although many quality management tools and techniques are being developed, in certain regions there is a negative trend implying that local organisations are not producing quality software. The authors found that QA practices were neglected and some of the practitioners were not aware of international standards. Some practices need to be taken seriously for quality software delivery. QM frameworks and tools are needed which require minimum time and cost.

Sommerville (2011) study stated that quality management at an organisational level needs to establish a framework of processes and standards to produce quality software products. The study illustrated how quality management could be used as an independent check in the software development process. However, the author did not explain the interaction between each development process in the product life-cycle and quality management process.

Aregbesola *et al*. (2011) identified key process areas that investigate software quality which are QA and QM practices in the Nigerian software industry. The study revealed a low performance in these areas. Kuhrmann, Diebold, Münch, Tell, Garousi, Felderer, Trektere, McCaffery, Linssen, Hanser and Prause (2017) carried out a hybrid approach to software development by combining waterfall and Scrum methodologies. The authors stress out the importance of customising processes to suit a particular organisation. Pandey *et al*. (2013) discussed a software agent framework to support software quality management. Chevers and Grant (2017) study stated that software organisations still struggle to deliver products that meet customers' needs. The study revealed the following to be some of the challenges that organisations: developers' skills and to a lesser extent process maturity and application of the latest technology. Other considerations were top management support, the industrial environment and team dynamics as other factors which can influence the quality and success of the delivered software product.

It is important to note that the body of knowledge needs new techniques on quality management approaches to iterative software development to achieve excellence (Alhassan *et al*., 2017).

To summaries related studies, most of them focused on ISD and software quality management in general. None of them focused on the inadequacy of QM in ISD or investigated the problem from a South African perspective.

2.7 Summary

In this chapter, literature related to the research problem was reviewed and discussed. The chapter began by presenting a general overview of common challenges, software development process approach that has historically and are currently being used. Next, software quality management attributes and standards were discussed. Related studies were also discussed, bringing out the gap which this study attempted to address.

The next chapter discusses the theoretical underpinning of this study by exploring structuration theory and duality of technology model as a lens through which to understand the software quality management problem and its institutionalisation.

CHAPTER THREE

THEORETICAL UNDERPINNINGS

3.1 Introduction

Given the problem statement and figure 1.3, there are underpinning sociotechnical processes in the interplay between the actors which, arguably, make quality management in software development projects a social phenomenon. As such the problem was studied through the lenses of a social theory to understand and interpret the embedded phenomenon. According to Harrington (2005) social theory can be defined as the scientific way of studying social interactions or social life. How societies change, develop and explain social behaviour and about power and, social structure.

The application of social theory has been widely used in IS research. It brings meaning to research by answering research questions from social-technical perspective. This study was studied by using socio-technical approaches found in the duality of technology (DoT) model of structuration theory by Giddens (1984). Duality of technology (for example, Information Technology) helps in understanding the relationship between ST and Technology (Orlikowski & Robey, 1991).

Structuration theory has provided theoretical frameworks for many empirical studies including business and informatics studies (Bryant, 1999). Structuration theory has been cited by many IS researchers which shows its popularity in this field (Jones & Karsten, 2008). Structuration theory in IS studies, has been used to rationalise the social phenomena associated with its processes and practices.

This chapter gives an overview of structuration theory, its key concepts, ST and technology, and goes deep into discussing DoT model, its key components and applicability to this study.

3.2 Structuration theory

3.2.1 Structuration theory overview

Structuration theory according to Poole and DeSanctis (2004) bridges two philosophies of functionalism and interpretivism, where the former gives power to structure and the later power of human agency. According to Albano, Masino and Maggi (2010) ST is considered important because it considers organisational dynamics. According to Hamman (2009) organisational change is about changing organisational systems through alteration of day-to-day practices

and the social structures by which they are governed, giving attention simultaneously to the individual parts and to the greater whole. Such an approach acknowledges that social practices activate social networks in which insignificant actions can have massive impacts. According to Wanyama and Zheng (2010) the structure and culture of an organisation does affect project outcomes. Structuration theory can be used to explain some of the meanings, norms and power issues experienced during software product delivery process. In this study, the implication of ST was used to inform and improve the quality management process in iterative software product development.

3.2.2 Structure

Structure according to Giddens (1984), is the structuring of attributes in social systems across time and space. Rose (1998) defined structure as rules and resources organised as objects of social systems such as practices, manuals, laws, and raw materials. These become rules and resources when incorporated within processes of structuration (Naidoo, 2009). Rules and resources are used by actors in interaction (Turner, 1986).

Rules refer to how to get things done routinely and are procedures of social interaction of meaning. People, as social actors possess knowledge gained from experience and apply this in their interaction in the social system. According to Turner (1986) for effective communication, rules provide a mandate which is the basis for sanctions and provides interpretive schemes and knowledge.

Resources refer to what agents rely on to get work done. It is the authority over people or things and rules which are guidelines for action which create a social system with power (structures of legitimation), norms (structures of dominations) and meanings (structures of signification). Resources through signification and legitimation are structured attributes of social systems, derived from and replicated by informed agency during interaction. Resources are the means through which power is exercised (Giddens, 1984). There are two types of resources, allocative and authoritative. The former involves command of objects and the later command over people. Allocative resources are raw materials and technology. Authoritative resources grant decision making power over the resources. Authoritative resources may be technical knowledge or authority (Burns, Englund & Gerdin, 2011). As Giddens explained people have more agency the more resources they hold and so can negotiate more rules (Whittington, 2010).

Daily structuration routine makes the social order of a system (Poole and DeSanctis, 2004). Structuration are the factors that enable how structures are transformed and progressed influencing the reproduction of social systems (Rose, 1998). Structuration is the process of executing structures. Changes in practices as well as regularities and continuation of structures (Poole & DeSanctis, 2004).

As depicted in figure 1.3 (conceptual framework) structure is quality management embedded in iterative software development process. Software practitioners as actors are sanctioned by these structural elements when developing the software products. Over time with each product iteration, through interaction with customers, management influence, architecture the quality management process will evolve.

3.2.3 Agency

Agents are individuals or groups of individuals derived from structures to do actions via memory traces (Gehman, 2008). Agents transform structures through reflexive monitoring and rationalisation. Structures are produced by the human agency through action drawn from knowledge of the system. In software product development collaboration is an essential aspect for agents through action (Timbrell, Delaney, Chan, Yue and Gable, 2005). Agency or action is a continuous flow of manner which make the daily activities of agents. Agency is the capacity to do work. Through activities, agents reproduce activities which make activities possible and human knowledge essential to action (Naidoo, 2009). According to Whittington (2010) agency is the capacity to do otherwise, to follow certain practices and refuse the other hence making up the design of a system. Agency relies on resources and agents can apply past experiences into current situations (Reflexivity). According to Giddens (1984) the rationalisation of action within the diversity of interaction is how the competence of actors is judged by others. Actors expect others to monitor the flow of their activities as they do. Actors monitor the flow of their activities and expect others to do the same. Human agency is defined in terms of intentions. Acts occur because the agent intends them but must be intentional under a description that another is wrong. It is in the capability of agent doing things which implies power. Agents in their power can act differently based on situation, and thus power should not be referred to as a resource. Action is determined by individual capability to make a difference. Naidoo (2009) argues that social structure is extracted by agents in daily activities, structure exist because of action and action is drawn from structure. Rose (1998) further argue that agency when connected to power can make a difference and lead to exploitation of resources driven by human action and the surroundings.

As figure 1.3 illustrated, with agency, agents can act in a way that can improve the software delivery process or comprise it. Agents are the developers, testers and product managers involved in the software development project. Agents will use their knowledge and expertise to make software product development decisions.

3.2.4 Duality of structure

Structure is an agency and product at the same time. A social structure has agents and are a product of past activities of those agents. Giddens stated duality and structure in addition to concepts of recursiveness as the core of ST (Gehman, 2008). Duality is two way, back and forth process where agents and structures are drawn upon from social systems which become part of duality. Actions produce and reproduce social structure. The social structure enables or constrain actions and social structures eventually becoming social systems. In examining social systems, ST examines structure, modality and interaction. Modality is how structures are translated into action (Twum-Darko and Iyamu, 2015). Agents through interaction can create and recreate a quality management system that will lead to the production of a quality product.

Structure and agency co-exist as duality of structure as structure constraints action and action modifying structure. Both structure and interaction are linked to modalities (Twum-Darko & Iyamu, 2015). The analytical dimensions of social practices are illustrated in figure 3.1 below:

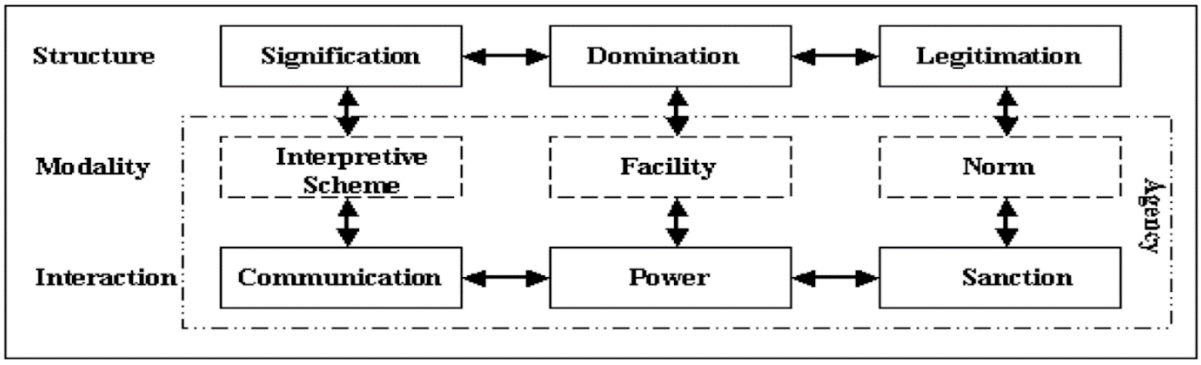


Figure 3.1: Dimensions of the duality of structure

(Adapted from Giddens, 1984)

Structure (determination) and agency (freedom) both play a critical role in influencing an individual (Oppong, 2014). According to Whittington (2010) agency is performed through communication, power and sanction with function of structural dimensions of systems - signification, domination and legitimisation.

Signification refers to a system's discursive and symbolic order – that is, rules governing the types of talk, jargon and image that predominate. Legitimisation refers to the control of norms

and sanctions; these rules extend from formal legal limitations and conditions to the kinds of unwritten codes that are embedded in an organisation's culture. The dimension of domination concerns material and allocative resources; these concern political and economic institutions, most obviously the state or the firm. It can be readily seen that these three dimensions connect structuration theory directly with issues of discourse, power and institutional legitimacy that are prominent throughout organisation and management theory (Whittington, 2010).

Modality is how structural dimensions are expressed in action. Modalities, as defined by Giddens are what actors derive meaning and social interaction. Modalities are both the media and output. Thus, in communicating, people draw on interpretive schemes that are connected to structures of signification; in exercising power, derived from what Giddens calls "facilities," for example, rights defined by the dimension of domination such as those pertaining to organisational position or ownership; and, in sanctioning, they draw on norms of appropriate behaviour embedded in the structures of legitimisation (Whittington, 2010). Meaning communication, power relation operations and sanctions happen simultaneously in social practices and interaction.

In this study, duality of technology has been used to understand how each product iteration is influenced by quality management process. All the people involved in the development process must adhere to some standards and follow certain practices. This will influence how social practices are produced and reproduced in the quality management process.

3.3 Structuration theory and information systems

3.3.1 Overview

Organisation information systems are organised as social structures which make up a system. According to Wanyama and Zheng (2010) norms practices and processes show that they are part of the socio-cultural and political-economic environment in which they occur and are influenced by that context.

Structuration theory appeal to software development focuses on structure and process by which structures are used and enhanced over time. The Information Systems (IS) field has a deep-seated concern with analysis and design of structures for decision making and human-computer interaction. Research focus has shifted to the structure of the interaction of people and technology and trying to improve processes (Poole & DeSanctis, 2004). Allison and Merali (2007) stated that actions are a flow events and structures keep changing because of agents'

actions meaning the defined processes in use that shape quality management in software development will not remain constant.

Hamman (2009) stated that alterations of current practices influence social systems to embed in those practices and according to Stark (2015) people play a major role because technologies are designed, developed and managed by people. Also, developers and others involved in the development process create and recreate the software products and processes (Cohen & College, 2014). These are not detached from human culture and according to Murgia *et al.* (2014), human emotions such as happiness and sadness can influence the effectiveness of collaboration amongst software product teams. Sedano *et al.* (2017), to add on stated that developing a software product is a complex socio-technical activity which involves people of different skill set and disciplines working together thus waste can emerge that may cause the dissatisfaction for the customer.

Giddens did not directly address ST application to IS; this birthed a version of ST called Adaptive Structuration Theory (AST) and the Duality of Technology (DoT). Both theories have sought to incorporate technology into a structuration framework. According to Jones and Karsten (2008), DoT is more favoured by IS researchers than AST because AST has little resemblance to the original ST principles (Jones, 1997). This study adopted the DoT model as a theory of analysis because of its applicability to the research phenomenon.

3.3.2 Duality of Technology

DeSanctis and Poole (1994) developed adaptive structuration theory (AST) to address the influence of technology and social theories. This theory focused on the concepts of appropriation and structuration. Jones and Kartsen (2008) on AST argue that social structures are frameworks which developers incorporate some of these into the technology with the result that structures may be reproduced or modified, creating new structures. Their argument brings into perspective the mutual inclusiveness of technology and action of technology users.

Orlikowski (1992) on the other hand took this debate further and developed a structuration model to explore the field study of information technology with the move to gain deeper insights of the relationship between technology and organisations. The understanding of this relationship Orlikowski stated that it helps in understanding the power of agents, the making and use of technology and design of organisations. Technology in organisations is a mediator of subjective human action within structural and cultural contexts, and this happens in dualism contributing to social structure (Orlikowski & Robey, 1991). There is constant interaction

between structure and agency through knowledge of human actors and technology is the enabler and constraint of such interaction through Giddens three modalities (interpretive schemes, norms and facilities) which Orlikowski (1992) stated its also reflexive and affects the social system.

Duality of technology model is a critique of Gidden's DoS, it emphasises the importance of power, meaning, norms and interpretive schemes. Technology assumed structural properties according to Orlikowski (1992) and stated that developers of software products are detached from it once its deployed and meaning derived from the human actors that created gets lost. An organisation's institutional properties affect human action regarding intentions, norms, design patterns and resources. In another study Orlikowski (2000) improves the DoT by describing technology as a medium used to enact structures, rather than embodied, via technologies-in-practice. How human action through interaction in ongoing practices enact structures through which technology is developed. Repeated use of technology artefacts which become institutionalised in organisation Orlikowski (2000) called this technology-in- practice a process of structuration (Jones & Karsten, 2008). Figure 3.2 below illustrates this process.

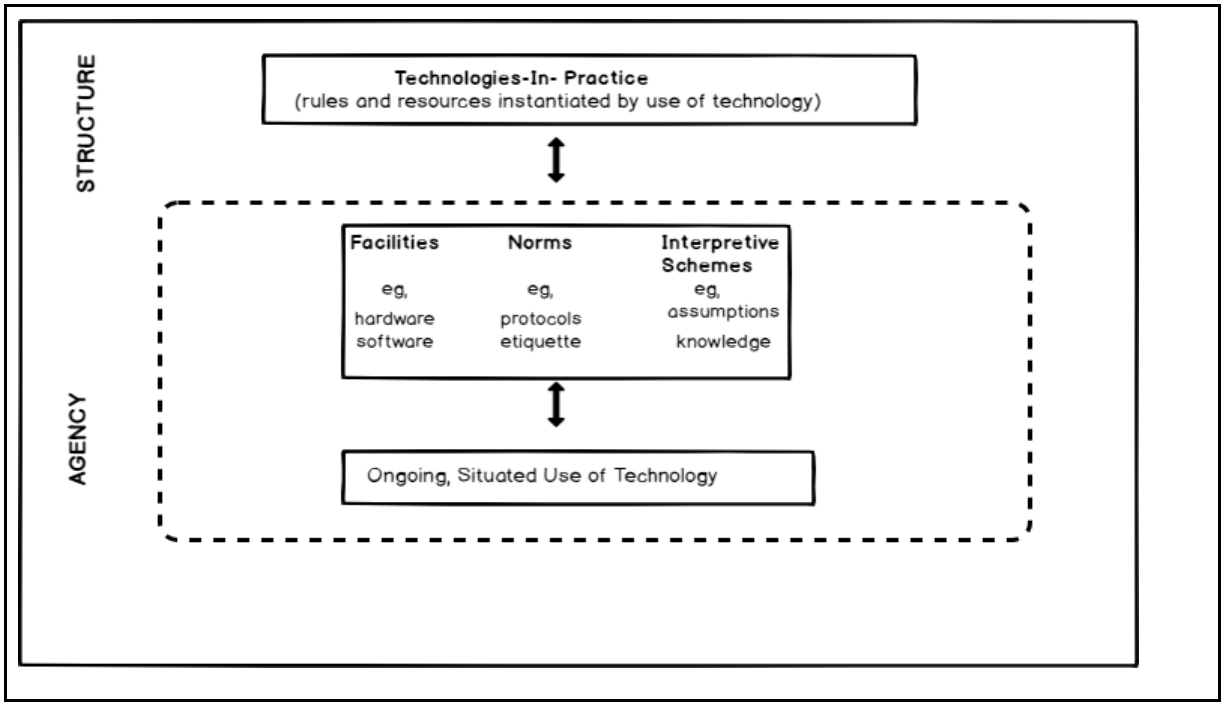


Figure 3.2: Enactment of Technologies - in - Practice

(Adapted from Orlikowski, 2000)

Enactment of technologies - in - practice represents how software development is unstable because requirements change. Generally, people are influenced by their environment. Hence

the need for iterative software development which is ideal for unstable environments. The quality management system needs to be aligned with the software development method.

According to Orlikowski (2000), people apply rules through knowledge by using hardware or software to complete their tasks. The dimension of the duality of structure of structuration theory illustrates the relationship between process, people and technology in an organisation. Given enactment of technology in practice, quality management in iterative software development process is embedded with communication, power and sanction for repetitive use/appropriation of technology.

Facilities are resources which drive outcome through power. Power according to Broger (2011) is associated with exercise as the human capacity to work for or against an organisation. Facilities when legitimised lead to structures of domination and through enactment with norms they make legitimised practices and sanctions if followed or not followed. Norms are necessary as they make quality management process effective. According to Twum-Darko (2014), norms are moral codes, leadership and skills for human interactions. Through communication, people use interpretive schemes to draw meaning of their interaction and interaction also creates interpretive schemes (Twum-Darko, 2014).

To summarise modalities and in the context of iterative software development and quality management, negotiated “norms” and “interpretive schemes” of software practitioners and customers (users of the software) translate to software products (Stillman, 2006). Software product development from a structural perspective is done through socio-technical processes resulting in structures embedded in the software development and the quality management processes. Software development in its context is a social practice which enacts structures and interaction through human actions and use of technology. Rose and Scheepers (2001) suggested a need for a framework to be adopted to improve development process regarding use of technology and social needs.

Loureiro-Koechlin (2008) stated that software development can be analysed as structuration processes. Software developers are act and enact structures in working environments and deployment of software products. They are also at the same time influenced by customer requirements, but they make their own interpretations of the requirements. Developers may have their own opinions about requirements what they feel is important to their business but might not exactly be customer requirements. As Giddens (1984) stated, human action is driven by their knowledge from past experiences of norms and facilities exposed to them. Habits enact structures of a social system (Loureiro- Koechlin, 2008). Software never stays stable

because of the interaction that happens with people making software and those using it. However, as software products entirely depend on technology, according to Orlikowski (1992), if there are any disruptions this may cause delays, increase costs for the organisation and customers are most likely to be affected and dissatisfied.

3.4 Summary

Technology does nothing except as implicated in the actions of human beings. Some researchers have criticised ST to be unsuitable for empirical research but rather just as a way of thinking that as an empirically testable explanation of social behaviour (Jones & Kartsen, 2008).

The purpose of this chapter was to explain the theoretical underpinning of this study. How quality management in software development can be analysed using theoretical lens. It explained social theory particularly ST in IS and DoT in detail as the key theory used to analyse results of this study. The importance of it in the understanding of how meanings are driven by structure and how structure influence actions and vice versa.

The concepts explained in this study influenced the following chapter and helped in drawing meanings in Chapter 5. The following chapter outlines and describes the research methodology.

CHAPTER FOUR

RESEARCH DESIGN AND METHODOLOGY

4.1 Introduction

The previous chapter gave an overview of structuration theory and its application to information systems research. It further went on to explain how DoT model can be used as a theoretical lens to analyse quality management in software product development.

This chapter discusses the philosophical assumptions, research design, methodology and methods for data collection and analysis. The research study is a qualitative case study, which used semi structured interviews to collect the research data. The main objective of this study was to investigate the influence of quality management process factors on the development of quality software products to address the problem of inadequate quality management in the software development process. The adoption of the research design and methodology helped to develop a better understanding of the phenomenon and attempted to address it.

The nature of the study and the study goal determine the research design. It is determined by the type of knowledge intended to be generated (Draper, 2004). Rowley (2002) defined a research design as a strategic framework used to conclude from the research questions. This study will use an empirical study featuring a case study to have deep understanding of the research problem by tapping into the experiences of the participants.

4.2 Research process

Appropriate research methods need to be selected to address the research problem. This study adopted Saunders' research onion in figure 4.1 below. The onion gives choices, paradigms, strategies and steps a researcher can follow (Mafuwane, 2012).

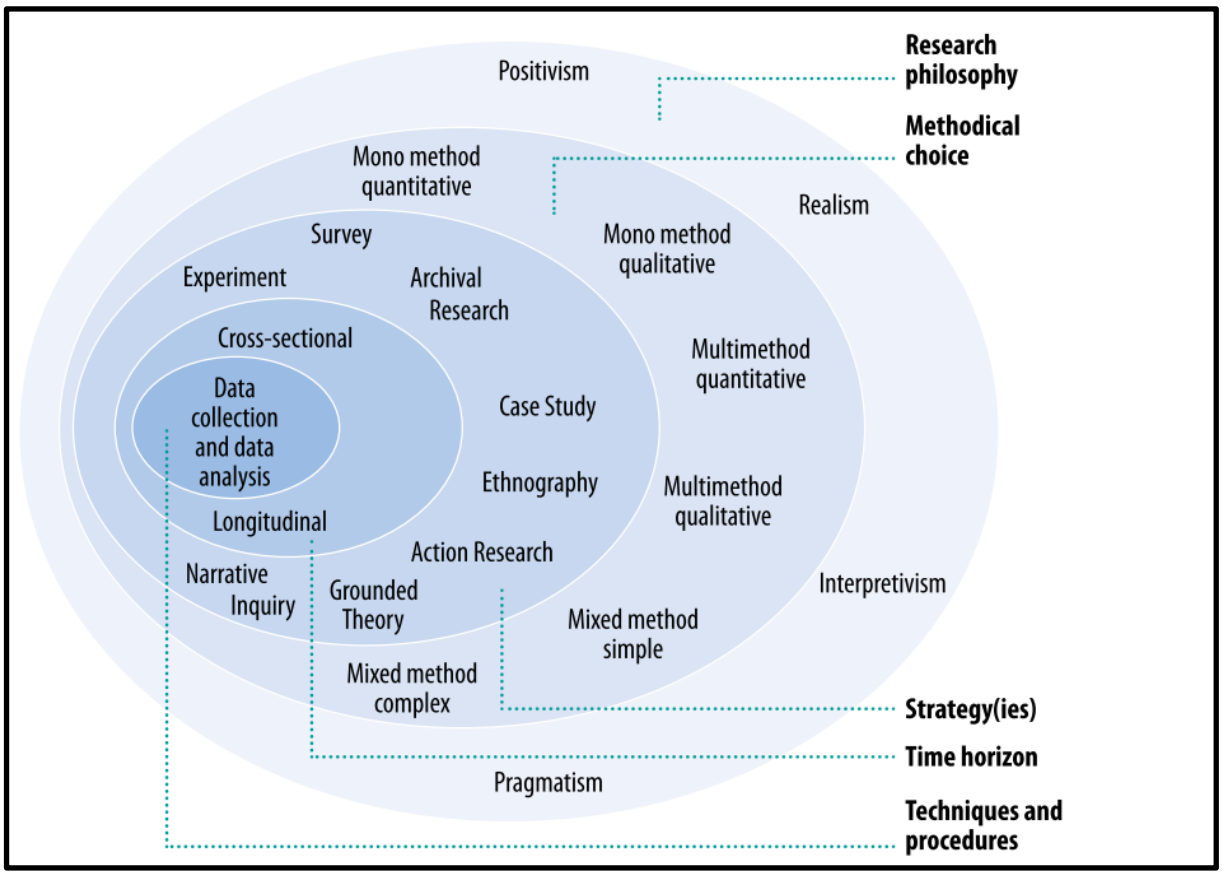


Figure 4.1: Research Onion

(Adapted from Saunders & Torsey, 2013)

4.3 Research philosophy

In research, there are some philosophical assumptions that need to be taken into consideration for it to be a valid study. Thus, it is important for these assumptions to be known (Thomas, 2010). A research philosophy according to Myeko (2014) is a way in which data is collected, analysed and used. Positivist (quantitative paradigm) and interpretivism (qualitative paradigm) are the two research philosophies.

Positivism is a theoretical phenomenon that brings objectivity and stability of reality (Munyaradzi, Maxmillan & Amanda, 2013). On the other hand, according to Gray (2013) interpretivism is based on social world culture and historical interpretations. The researcher is involved in the social context (Thomas, 2010). Interpretivism asserts the natural reality and social reality require different methods since natural science is based on data consistencies to draw patterns and to put systems in place while social sciences are based on the activities of the individuals. The key to interpretive philosophy is carrying out a study in its natural environment and acknowledging the possibility that the researcher cannot avoid affecting the phenomena they are studying (Munyaradzi *et al*., 2013). Interpretivists assume that knowledge and meaning are acts of interpretation, therefore no objectivity (Thomas, 2010).

Carlsson (2006) stated that most IS studies adopt a positivist approach which arguably is not always the case (Mkansi & Acheampong, 2012). Munyaradzi *et al*. (2013) concur by stating that perhaps the reason why there has been difficulties in IS research is that of the inappropriateness of positivist approach.

The purpose of this study was to understand software quality management process for SasTech. To address the research phenomenon the study required extensive descriptions and explanations thus it followed an interpretive philosophy stance which according to Mora *et al*. (2012) provides meaning to complex realities and provides in-depth insights into the context of the organisation. The interpretive approach was achieved through conducting interviews with SasTech software product development team.

4.4 Research methodology

Inadequate quality management in iterative software development process has not been studied adequately leading to this research study. A qualitative phenomenological approach was used because it derives meaning from personal knowledge and experience of the participants (Lewis, 2015). This subsection discusses the research method used to present new knowledge based on empirical affirmation and logical evidence. Research methodology is a way which determines if certain issues can be researched. Research methods refer to ways in which data can be collected (Johannesson & Perjons, 2014). There are three methods which can be applied to research which are qualitative, quantitative or a mix of both which is termed mixed methods. According to Venkatish, Brown and Sullivan (2016), mixed methods combines quantitative and qualitative research methods in the same study to gain deeper insights and counteract weaknesses of both methods if used separately.

Quantitative research is numerical, non-descriptive and applies statistics or mathematics. The process is iterative, described in quantities and expressed in numbers. Tables and graphs are used to present the results. Qualitative research investigates the what, where and when decision making and is conclusive (Rajasekar, Philominathan & Chinnathambi, 2006). Quantitative research uses a questionnaire as a data collection tool. “A questionnaire is a form containing a set of questions, especially addressed to a statistically significant number of subjects, and is a way of gathering information for a survey. It is used to collect statistical information or opinions about people” (Mafuwane, 2012). A questionnaire can consist of both open and close-ended questions. The participants must answer the questions on their own time which may take long to get responses and allow for other questions to be skipped because participants do not understand what is being asked.

Qualitative research involves the study of phenomena in its natural setting to draw meaning. The rationale behind qualitative research is that it explores individual experiences (Nicholls, 2017). It draws meaning from people's words and actions. Qualitative study also involves phenomena to do with quality whereas quantitative focuses on the quantity to derive the phenomenon (Kothari, 2004). Nicholls (2017) also argued that qualitative is more real-life focus for an interest which has not fully been studied. This study aimed at exploring the quality management factors in software delivery at SasTech therefore, an in-depth investigation in a qualitative study rather than quantitative was needed to understand the research phenomenon. On this basis, qualitative methods were adopted and applied to the research study.

4.5 Qualitative research methods

According to Myers (1997), a research method is an approach that a researcher takes from philosophical assumptions, moving on to research design and then data collection. Whatever research method is chosen affects data collection and may require different skills assumptions and practices.

The common methods in IS research are action research, case study, ethnography and grounded theory. In action research, the researcher and participants work together, get results that drive further action closely to examine their own and knowledge gains drive further action. It is done verbally rather than through numbers and the emphasis is put on transformation in social settings (DeVilliers, 2012). Grounded theory uses themes, patterns, and theories which emerge inductively from extensive data collection, coding, and analysis of the research data (DeVilliers, 2012). Ethnography method, research is done in its natural setting over an extensive period through patterns of behaviours, language, and actions (Cresswell, 2013). Case study research according to Zainal (2007) allows data to be examined within specific conditions. It helps explore real world environment explain the complexities of the real-world situations which may not be captured by other methods. Rowley (2002) defined a case study as an empirical investigation of reality. It is an in-depth study of a complex phenomenon usually of a single or multiple unit with the intention to generalise the whole (Gerring, 2004; Baxter & Jack, 2008).

4.6 Overview of the case study

The case study concerns a South African software development organisation (SasTech, fictitious name). SasTech was founded in 2015 in Cape Town, South Africa but has since

opened other offices in Durban and Johannesburg. It has developed many virtual products for both businesses and consumers. These are accessed via mobile apps (Android and iOS) and websites. The organisation has a national customer base and has over 100 employees, however only 24 employees are involved in the software product development process. The employees involved are organised into product management team reporting to the chief executive officer, quality assurance team currently reporting to the development manager and software development team reporting to the development manager.

Although this study will focus on SasTech processes, Sommerville (2011) note that the need for an effective quality management process is universal. A case study is suitable for this research since it allows a better study of complex issues and enables the researcher to explore data in a specific situation. The case study approach beyond qualitative study allows researchers to tap into the unknown and advance knowledge in known areas as well (Starman, 2013).

4.7 The data collection method

Data collection begins when a research problem has been defined, and research design has been determined. The researcher should keep in mind whether they want to collect primary or secondary data. Primary data is fresh from the source and secondary data is that which has been collected and processed by other researchers (Khotari, 2004). Quality management factors and challenges in software delivery are related to socio-technical issues which impact behaviours which made qualitative primary data collection methods suitable for this study. The main methods are interviews, focus groups, observation and questionnaires (Hancock, Ockleford & Windridge (2009).

Focus groups employ topic guides to help keep discussion relevant to the research. They can be more difficult manage and convene because many people are being interviewed at the same time but can be used to analyse how participants influence each other (Hancock *et al*., 2009).

Observation is used when observing participants is more valid in a research setting to determine behaviour instead of what is being said (Hancock *et al*., 2009). This method is suitable in a setting where participants cannot freely provide verbal feedback. However, a researcher is likely to get limited information and can be very expensive and is dependent on the accessibility of the participants (Khotari, 2004).

Interviews have been claimed to be more suitable for qualitative research (Khotari, 2004). According to Isaacs (2014), the most common qualitative method are interviews. Interviews are used to explore views, experiences, belief, understandings, knowledge and motivations of people under study (Frederick, 2013). Interviews can be in the form of structured, semi-structured and unstructured (Gill, Stewart, Treasure & Chadwick, 2008). According to Hove and Anda (2005), in software development semi-structured interviews are ideal but, caution needs to be taken because they are costly, and quality of the research is dependent on how the research is conducted. Thus, planning is critical to obtain the best outcome for the research.

The objective of this study was to collect data from participants regarding quality management in an iterative software development process at SasTech. Therefore, the interview method with proper planning was chosen as the data collection method for this study because of its suitability for a qualitative study and the need to collect data in detail from the participants. Semi-structured face to face interviews were used to provide guidelines to participants. In semi-structured interviews questions were asked that allowed certain themes that emerged from the participant's answers to be explored further unlike structured interviews which do not allow follow up questions.

4.8 Sampling design

Sampling is selecting some part of the population to represent the whole population (Thompson, 2012). Before the data was collected, the study used a sample design that followed the process of defining the population, determining the sampling frame, selecting the sampling technique, determining the sample size and executing the sampling process to select the sample unit (Khotari, 2014).

4.8.1 Population under study

Drew, Hardman and Hosp (2007) defined population as "all constituents of any clearly described group of people, events, or objects who are the focus of the investigations." According to Khotari (2004), population can be limited or unlimited.

The population for this study was finite. It comprised of 24 people in quality assurance, product management and development teams. The participants were identified from these teams with the organisation's consent to interview them (see appendix A).

4.8.2 Sample frame

A sample frame is a list of individuals who have a chance to participate in a study (Fowler, 2002; Khotari, 2004). The sample frame of this study included all the teams at SasTech involved in software development. Every individual in these teams was eligible for selection. The sample frame was identical to the population.

4.8.3 Sampling methods

Non-probability purposive sampling was used because the sample representative was based on the knowledge of the phenomenon and the purpose for which the date was being collected (Barreiro and Albandoz, 2001). The purpose of this study was to get rich insights into factors that affect quality management in the software development process. Drawing from Palinkas, Horwitz, Green, Wisdom, Duan and Hoagwood (2015) purposive sampling was considered because of the availability, willingness, experience and the ability of the sample representative to communicate effectively on the phenomenon being addressed.

4.8.4 Sample size

The sample frame consisted of 24 participants. The projected sample size to answer the research questions comprised of 21 participants. According to Isaacs (2014), there is no formula in a qualitative study, 12-26 people are considered fair. However, the sample study size can increase if necessary during data collection.

Furthermore, the study was conducted with the aim of upholding a margin of error of 5% with a confidence level of 95% and a 50% response distribution. 21 participants were ideal. All 24 participants were asked to participate, but only 22 participated. The other participants had no time to be interviewed on the agreed date and time. Participation was based on willingness.

4.9 Interview process

The nature of this research required a data collection method that is flexible and to capture people's views to make meaning of their knowledge and experience. A process was to be followed to choose the type of interview and in this case, semi-structured interviews, ethical considerations, defining the interview protocol, conducting and recording the interview and findings reporting (Rabionet, 2011).

The interview process aimed to address the following research questions:

(a) How is quality management process in software product development influenced?
(b) How are socio-technical processes embedded in the quality management process of software product development?
(c) Why are software development processes ineffective?
(d) How can quality management in software product development be guided to meet customer expectations?

A list of semi-structured interview questions was designed (see appendix C) to cover the core topics of inquiry to provide content and to provide direction important to the study.

4.9.1 Pilot study

A pilot study was done to pre-test the interview questions. Three (3) interviews were conducted with participants from another organisation. Data was collected in September 2017. The final interview guide was then submitted for ethical clearance at Cape Peninsula University, and ethical clearance was granted (see appendix B).

4.9.2 Actual interviews

After ethics approval was received, the actual interviews were conducted in December 2017 and January 2018. Participants were verbally asked to participate in the interviews with scheduled times. Questions were shuffled depending on how the interview was proceeding. The interviews were with practitioners involved in the software development process.

4.9.3 Data Recording

Field notes of important information taken in this study were collected from the participants using a digital audio recorder with the participant's consent. All the recordings and notes were kept in a spreadsheet for analysis at a later stage. Each participant was assigned a pseudonym. The pseudonym was used for everything relating to that participant. To minimise bias, a relaxed environment was chosen where the participant was able to freely express themselves. The actual data collection was informed by the sampling process to collect the correct data needed.

4.10 Qualitative data analysis

Data analysis is the process of analysing data to interpret the research results and form conclusions (Theron, 2015). It is what researchers do to make sense of the transcripts. Data

was validated after each interview so that context was not lost. The complete analysis was done after the study was finished. According to Vosloo (2014) data analysis techniques are determined by the research objectives and questions.

The process outlined by Aronson (1995) and Isaacs (2014) was also followed. These are:

- Reading and re-reading the interview transcripts and listening to the recordings to make sense of what was said and going back and forth between data, study aim and theoretical framework (Isaacs, 2014);
- Data analysis through thematic analysis. Thematic analysis according to Boyatzis (1998) is a process for encoding qualitative data by identifying and analysing the results and organising them (Isaacs, 2014);
- Draw patterns of experience, paraphrasing like ideas or direct quotes;
- Identify all the data that relate to the pre-existing classified patterns. In this case, predetermined themes from DoT were used, and for discussion of the study, themes were derived from patterns that emerged from the data;
- Related patterns are combined into sub-themes;
- Build an argument for choosing the themes that validate choosing those themes by reading related literature; and
- Development of storylines from the themes.

4.11 Data reporting

Tables were used to present the data to ensure the flow of data can be followed.

4.12 Reliability and validity

The reliability of this study was ensured by how the study was carried out in line with qualitative research techniques and full consent of the participants (Vosloo, 2014). The three checks according to Vosloo (2014) to ensure reliability and validity of qualitative data are:

Credibility: Use of transcripts and notes to conclude from the data.

Authenticity: Iteration of interview question through a pilot study to reduce and validation of data.

Confirmation: Auditing of data to draw principles and trends drawn from the data.

4.13 Ethical Considerations

There were privacy and confidentiality concerns regarding disposal of personal or organisational data. Therefore, to address the ethical concerns of both the respondents and the University, full consent was obtained from SasTech and CPUT before the study. Privacy and anonymity of individuals and the organisation participating in the research were protected. Any type of communication about the research was done with honesty and transparency and participation was voluntary.

4.14 Limitations and delineation

This research was a case study of one organisation in Cape Town, South Africa. Therefore, findings may not be generalised for all organisations, but this will provide a general framework which will consequently make a valuable contribution. The intent was to reduce the size of the sample but to provide a much-detailed analysis to understand the phenomenon and then conceptualising the findings of which future studies can be built on. The main factors driving the decision to limit the size of the study area was the available time and resources as the study must be completed within two years under a limited budget.

4.15 Summary

Chapter four discussed the research methodology used in the study. A qualitative case study approach was used. 22 semi- structured interviews were conducted with the three relevant teams at SasTech. The following chapter, Chapter Five presents and discusses the findings of the study.

CHAPTER FIVE

FINDINGS AND INTERPRETATION

5.1 Introduction

The last chapter discussed the theoretical assumptions, research design, methodology and methods for data collection and analysis. Following a qualitative case study approach, data was collected, and results were drawn.

This chapter, therefore presents the analyses, descriptions and interpretations of the results in a systematic manner based on the literature review outlined in chapter two and themes drawn from duality of technology model of structuration theory outlined in chapter three.

5.2 Unit of Analysis

Given Table 5.1 below illustrates the object of analysis where 24 participants at SasTech were selected, only 2 did not participate in the study. The outcome is explained in the subsequent sections.

Table 5.1: Field work

Role	Selected Participants	Participants who responded
Product manager	4	4
Designer	1	1
Developer	16	14
Manager	1	1
Quality Assurance (Tester)	2	2
Total	24	22

5.3 Data Presentation

The research problem was defined as inadequate quality management in software product development. Therefore, the aim was to explore the factors influencing quality management in software development process based on a case study, a software development organisation in Cape Town, South Africa. To address the aim, the main objective was:

"to investigate the factors influencing quality management process on the development of quality software products."

To answer the research question, three themes were drawn from the data using thematic analysis which emphasizes organisation and rich description of the research data. These themes are presented in this section; factors which influence quality management, quality management challenges and quality management improvements in ISD.

5.3.1 Theme 1: Factors influencing quality management in software product development

The aim of this theme was to understand the factors that influence quality management in iterative software product development based on how the software practitioners at SasTech understand quality management, how quality management is embedded in the software development process, socio-technical factors and what challenges they currently face.

Table 5.2: Factors influencing quality management in iterative software development

Factors	Number of participants
Documentation of correct requirements	3
Design	3
Planning	4
Development process and technology	12
Testing	4
Release process	8
Management support	5
Maintenance support and upgrades	1
Usability and user experience	2
Timelines	2
Process ownership	1

Factors influencing quality management were drawn and summarised from the participant's responses under the themes explained in Table 5.2 above.

Quality management was described by participants as:

What the tech team build to match what is required and is only as good as requirements put on it." (AJ-developer manager)

A systematic and manual process of determining whether a product or service meets specified requirements of a use case and specification. (TP-product manager).

*Quality management starts from the get-go when a product gets new requirements. (*BC- senior tester)

The literature reviewed provided a general idea which was that an effective quality management process leads to a quality product that satisfies the customer.

5.3.2 Theme 2: Quality management challenges in software product development

Challenges were categorised into five themes: planning, management, documentation, personnel and architecture and tools.

Table 5.3: Planning challenges

Challenges	Number of participants
Poor planning: Pressure to release and tight deadlines which tend to sacrifice time to test	1
Version upgrades that introduce side effects	1
Not anticipating all the use cases	1
Poor design	2

Planning challenges illustrated in Table 5.3 above associated with pressure to release quickly, poor design and incomplete requirements.

Table 5.4: Management challenges

Challenges	Number of participants
Resource limitations	5
Delayed releases	1
One man show syndrome	1
Lack of training	1
Developer lack of knowledge of all systems	1
Sprint meeting – one person represents the whole team	1
Lack of coding standards	1
Not enough of communication	1
No guideline, non-adherence to industry best practices- no formal rules	1
Pre-existing bugs management	2
Prioritisation politics	2
No quality assurance lead	4
Lack of expertise	2
Lack of a quality assurance process	2

Management challenges indicated in Table 5.4 include resource limitations delaying the release because there is not enough time to test. In addition, the issue of having one person working on a patch and only them aware of what is going on and having one person representing the whole team in meetings. There was also lack of training and no guideline or formal process on best practices.

Table 5.5: Documentation challenges

Challenges	Number of participants
No documentation - verbal requirements	1
Incomplete or poorly articulated requirements	4

The documentation challenges drawn from the results as indicated in Table 5.5 above were no documentation at all and requirements being verbally communicated to developers by product managers as well as incomplete and poorly articulated requirements.

Table 5.6: Personnel challenges

Challenges	Number of participants
The blame game between developers and QA	2
Developer mistakes, lack of code reviews *(a case of release and forget)*	2
QA team cannot catch all bugs	2
Older team members reluctant to adopt change	1
Poor collaboration	4
Waterfall tendencies – work delivered at one go	2
Developers working from home- not productive management need to control/ employee discipline	2
Lack of understanding of requirements	1
Pressure to meet “spaghetti” requirements	1

Table 5.6 above highlights all the personnel challenges. The most common challenges were poor collaboration, lack of skills and lack of thoroughness when personnel accomplish their tasks.

Table 5.7: Architecture/Tools challenges

Challenges	Number of participants
Lack of automated testing tools	1
Lack of QA tools	3
Legacy systems too old for automation	1
Third party system changes	1
Too many dependencies due to the interconnectedness of the system	4

Legacy issues, too many dependencies of the systems and lack of quality assurance tools were the architecture/ tools challenges highlighted in Table 5.7 above.

Tables 5.3, 5.4, 5.5, 5.6 and 5.7 above summarised all the quality management challenges currently experienced at SasTech. As described by one participant:

> *Creating a process in the first place is important to an effective process, right now, I feel there is no proper process in place, there is no proper planning, it is always adhoc stuff, with QA it is always can you do this or that.* (JK-tester)

From the results, most participants pointed out that the current process is not effective, and few said the process was good, but there was a need for improvement:

> *We are agile and this organisation like any other we try to make it suit our environment.* (BG-senior developer).

All the top challenges are summarised in Table 5.8 below:

Table 5.8: Top challenges

Challenges	**Number of participants**
Resource limitations	5
No quality assurance leader	4
Incomplete or poorly articulated requirements	4
Poor collaboration	4
Lack of QA tools	3
Too many dependencies due to the interconnectedness of the system	4

5.3.3 Theme 3: Quality management improvements in software product development

The following improvement categories were drawn from the participants' responses.

Table 5.9: Documentation improvements

Improvements	**Number of participants**
Clear and concise documentation: diagrams- architectural and sequential	4
Complete requirements from product managers	3

Table 5.10: Planning and management improvements

Improvements	Number of participants
Having a clear and concise process flow	1
Continuing education programs to upskill	3
Unrushed implementation or development	1
Cross-training	1
Hire people with expertise	3
Frequent meetings between sprints to touch base - retrospectives	4
Product manager host seminars with customers to elicit requirements - call centre	2
Better prioritisation- critical bugs; work tasks; backlog grooming, backlog with well-defined user stories	3
Better project management tool like Jira	1
Start over and redo legacy systems where possible	1
Moratorium/code freeze	1
Making time for quality management planning	5
Communicating on workload and timelines in advance	3
Stakeholder consensus	2

Table 5.11: Design improvements

Improvements	Number of participants
Clear and concise documentation: diagrams- architectural and sequential	2

Table 5.12: Development improvements

Improvements	Number of participants
Ensuring optimal error handling/trapping in the source code	1
Adopt Spotify engineering culture- creating autonomous teams	1
Unit testing	3
Peer reviews for efficiency and best practices, for example, pair programming	2
Code reviews	4
Technical testing - developers must test code before handing over to QA	5

Table 5.13: Testing improvements

Improvements	Number of participants
Fully test plan with pre-defined outcomes	2
Test script/ procedure outlining how to conduct tests	2
Making a checklist of all use cases and taking time of thinking of the edge cases	1
UAT before release	9
Pilot testing	2
Beta release	1
Automated testing, the recommended tool is Selenium	5
Risk-based testing	1
Testing edge cases	2
Load and stress testing	2

Tables 5.9, 5.10, 5.11, 5.12 and 5.13 above highlights the practices that can help improve the process. These are categorised under planning and management, design, documentation and development. Among these are, better planning, more QA resources, better collaboration and communication and well specified and articulated requirements and tools were the main improvements that the participants said would improve the quality management process in software product development. As explained by one participant:

> *Like any software organisation out there, there is always room for improvement.* (BG- senior developer).

> SasTech is heading towards the right direction. One of the participants stated:

> *About three years ago, there was not even a process or a QA team in place, but things are improving we are putting the infrastructure in place.* (MJ-UX designer)

Planning, collaboration, testing techniques, automation, technologies, customer involvement and frequent communication within teams were key improvements which were common in the transcripts.

5.4 Analysis and Interpretation

Orlikowski's duality of technology model (DoT) and particular enactment of technologies-in-practice was used to further analyse and interpret the data. According to Orlikowski (1992), technology is an enabler and constraint to agent action and is also a social product of agent action. Software practitioners are influenced by structures within their department at the same time by customer requirements and market dynamics. If structures are not enforced consistently or actors choose to ignore the structure that is intended to provide rules and resources, their actions will affect the structure.

Figure 5.1 below underpin the use of the theory for interpretation for this study. These components were grouped under technologies-in-practice Agency and Structure attributes. The general objective is to adopt a general framework that will guide quality management in ISD. The answers to the research questions led to the modification of the conceptual framework using DoT to be discussed in section 5.6.

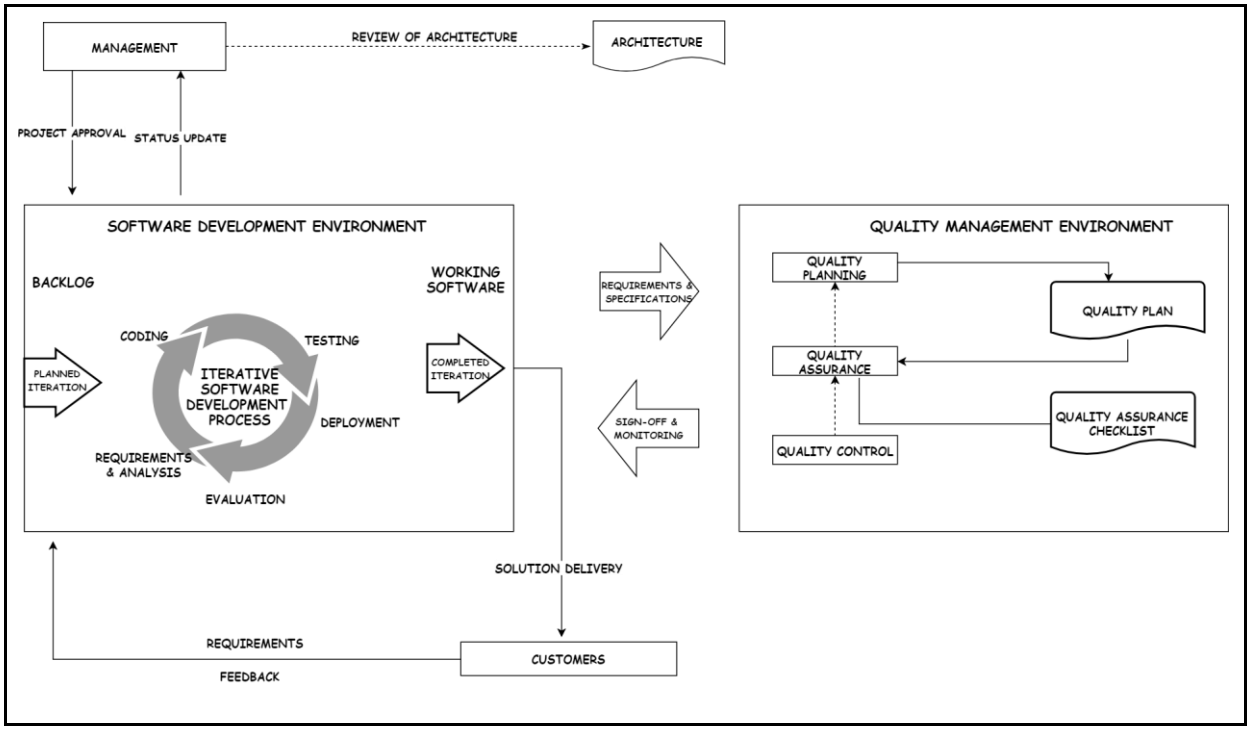


Figure 5.1: Conceptual framework

This study investigated the structures that enact quality management process in iterative software product development. The investigation was done through the three modalities of structuration – facilities, norms and interpretive schemes to get better insights into how quality management processes are embedded in iterative software development. Quality management process involves interaction among actors with different roles, professional background, skills, knowledge and expertise in charge of the delivery of the software product.

The organisation aims to achieve operational excellence by transforming and standardising the quality management process. The study used DoT to analyse the research problem of inadequate quality management in iterative software development and how it can be solved.

5.4.1 Human and non-human resources as Facilities

According to Timbrell *et.al* (2005) facilities are used in the exercise of power through domination. Facilities are human and non-human resources that power the creation of software products. Actions are guided by facilities available to actors (Loureiro-Koechlin, 2008). People build into technology certain facilities which are resources to accomplish work (Orlikowski, 1992).

Facilities produce structures of domination and are resources needed to produce a quality product. A facility or resource become a dominant factor based on how agents interpret it and use it. It is the means to be used to accomplish goals. Each person assigned to do a certain task has some level of power based on their skills and knowledge, so they influence the outcome of the product (Twum-Darko, 2014).

5.4.1.1 Human facilities

Quality management as a structure is constituted by human agents. Human facilities are enablers that make things happen through structures of domination. As people get hired by an organisation, they are influenced by the current process simultaneously the process is also influenced by their behaviour. These are product managers or product owners, quality assurance personnel or testers, software developers, customers, other business stakeholders. All these need to work together.

As put by one of the participants:

> *To ensure that everyone involved is in sync, all business and tech stakeholders meet every two weeks, and customers are engaged regularly, to ensure that their needs are constantly met*. (TK- senior developer)

Each role plays an important part in the quality management process to ensure that a quality product is delivered. Product managers own the product, they elicit requirements from the customer and business stakeholders and sign off each iteration once QA has passed it.

Quality assurance team is there to ensure that what has been developed is as per the specified requirements. Developers' role is to develop and deploy each product iteration.

5.4.1.2 Non-human facilities

Technologies are enablers to tasks being done. According to one participant:

> *It is important to adopt the necessary tools and software to keep up with tech. (BC-tester)*

There are various web and app technologies mentioned by participants in the study that enable them to do their work. These include new programming languages, integrated development environments (IDEs), version control systems, interfaces, testing tools, project management software and other third-party software systems.

5.4.2 Practices and routines as Norms

Norms refer to rules for understanding how to act, thus are used as standards to sanction or legitimise actions. These are structures of signification. Norms inform ongoing practices (Naidoo, 2009). Actions are guided by knowledge of norms (Mahdavi & Daryae, 2015). People build into technology certain norms which are rules that define how to accomplish work (Orlikowski, 1992).

Rules and regulations embedded in the process are legitimated by actions (Twum-Darko and Iyamu, 2015). Projects have an executive sponsor who legitimises the project but has certain processes that needs to be followed. These are norms. Legitimation is done through structure towards achieving a quality product.

Quality management in ISD is norms driven. Through daily practices and routines, they become a medium for quality management structure. Several key practices have been mentioned throughout the transcript.

Most participants when asked about quality management interpreted it as quality assurance (QA) owned by the quality assurance team or testers. In line with what quality management really is, one developer stated that:

> *QA just do quality assessment; the product managers have to take ownership because if they do not some the functionality might be missed by QA and that might lead to a poor-quality product. If it is my product, I make sure I test thoroughly.* (TO-product manager)

Product management role is key in this process because it is the intermediary practice between customers and the organisation. Also, within the organisation between the development, testers and the rest of the stakeholders in the organisation. Product managers report progress to executives.

Testing is another of the key practices involved in the quality assurance stage in the quality management process which was quite popular with most of the participants. The key functionality of testing happens when developers finish their work and hand it over to the QA team. The key practices mentioned in the transcripts are consistent, in-depth testing, developers testing their own code (unit testing), regression testing, device testing, testing edge cases, writing test cases, debugging when there are issues, code reviews, user acceptance testing (UAT). In addition, automated and manual testing. Currently at SasTech, code reviews, unit testing and automated testing are not practiced. These eliminate developer mistakes and allow bugs which could have been missed by QA to be picked up before release. Testers receive specifications from product managers:

> *We build test cases and get outcomes of the tests. If we pass we send to product managers to sign-off, if there are bugs we send the work back to developers to fix*. (BC-tester)

In general, as explained by one of the testers:

> *Rules give us a clear guideline of what needs to be tested in the beginning. QA needs to be involved throughout the whole software development life cycle, and testing is the responsibility of everyone involved. (BJ-developer)*

As standards of practice, coding standards and naming conventions were also mentioned in the transcript. One of the participants stated:

> *I apply ISO to my development Java card 7816, it's been built within Java to follow standards. (TJ-developer)*

Another practice popular is the meetings. The only official meeting mentioned was the sprint meeting which occurs on a two-week cycle. As a standard of practice regular meetings are recommended for alignment especially retrospective meeting recommended by Scrum.

5.4.3 Standard procedures and artefacts as interpretive schemes

Interpretive schemes are known as stocks of knowledge derived for actors' experience (Twum-Darko, 2014). They are the rules an actor needs to know. People build into technology interpretive schemes which are rules embedded with knowledge as to how to use the

technology (Orlikowski, 1992). Through interpretive schemes current structures are shaped and determine ongoing processes (Naidoo, 2009). Actors derive interpretive schemes from technological artefacts. These can be skills, power and emotional and intellectual abilities (Orlikowski, 2000).

How an event should be in interpreted? What is significant? The outcome of any project depends on how significant it is in the organisation which depends on how requirements, rules and regulations of the project are interpreted by the people involved in the project.

On skills, one participant stated:

> *Skills and experience of developers will ensure there are less rules. (RG-developer)*

In addition, product managers' ability to provide well specified requirements is also mentioned as key in driving lines of communication.

Sometimes there is poor judgement and software is deployed to production with issues.

> *There are always bugs, I would say from my experience 10-20% usually incorrect responses or UI related. (FD-developer)*

In addition, as explained by another participant:

> *There are issues that have been escalated to developers found in production. Oftentimes developers do not acknowledge their mistakes and blame other parties. (TK-developer)*

As a standard procedure for quality management, ISO 9001 as mentioned in Chapter One is a standard that can be adopted by an organisation. Most of the participants said they knew about ISO standards but had no experience. Only two of the developers said they had experience with ISO standards but not ISO 9001.

> *I know there are ISO standards specific to an area of software, e.g. testing, communication, depending on the type of software. So, there are many ISO standards, e.g. Postilion, a software I worked on before I joined here used ISO8583. Here we developed our custom standard. (TJ-developer)*

5.5 Institutionalising quality management process in software product development

Through facilities, norms and interpretive schemes technologies in practice are enacted. Through an iterative process, technologies in practice become structure, and it goes on and

on (Orlikowski, 2000). As actors enact altered technologies in practice facilities, norms and interpretive schemes change in the development and use of technology (Orlikowski, 2000).

According to Orlikowski and Robey (1991), software products are interpretive schemes where human action is translated to routines. Each iteration recreates structures of meaning that changes user behaviour. The development of software product is influenced by the current state of knowledge available, methodologies, time budget resources management objectives and organisational culture (Orlikowski & Robey, 1991).

5.5.1 Technologies and human resources as Facilities

Architecture, tools, enough resources with better planning and management are important to have an effective quality management process in ISD. Facilities as human and non-human resources such as architecture enact organisational structures of domination and people can change any structure of domination based on allocative and authoritative resources. Developers, testers and product managers can influence the management and structure of domination changes. Resources available in the organisation influence the goals that are set which need to be accomplished and power is exercised. Quality assurance tools, new programming languages such as Go, and Scala may contribute to delivery of a quality product. Lack of a QA lead, for example, has led to having a poor QA process. Resources can be modified when challenged and from an institutional view resources are structural properties that constitute organisational structures of domination (Orlikowski & Robey, 1991).

5.5.2 Best practices and standards as Norms

Organisations are guided by normative sanctions (Orlikowski, 1992) which provide guidelines on how work should be done. Structures of legitimation which form the organisation culture. In this case a quality management system with standardised work procedures. When a quality management system has been defined, over time it can become institutionalised. A sanctioned way of doing work is established and will become a way which may drive change in the organisation.

Quality management is thus defined as the: *standard definition of practice* (LI-product manager) the *standards by which quality is measured*. (RG-developer).

The continuous following of the standard procedures and best practices in quality management will see SasTech producing quality products with each iteration. If there is non-compliance will see the product diminishing leading to poor quality products and hence loss of business and revenue at SasTech.

The best practices include following coding standards, naming conventions, proper testing procedures and adopting what is relevant from ISO 9001. A change control form signed by all relevant parties before software release is another practice that may eliminate issues post deployment.

5.5.3 Architecture as Interpretive Schemes

The communication process is informed by social rules from structures of signification derived from interpretive schemes based on experience and continuous education brings. These stocks of knowledge will lead to creation of institutional elements of a quality management system. Best practices will need to be properly communicated through regular meetings, collaboration and peer reviews which transfers knowledge.

5.6 General framework of quality management -in-practice

Based on results drawn from this study and the conceptual framework, a quality management process guideline can be proposed. This is illustrated in figure 5.2 below. All processes which create a quality management system need to be created and recreated over time through evaluation which will lead to adequate quality management in ISD. Technology skills, power, knowledge, assumptions and expectations are acquired through continuous education, communication and experience (Orlikowski, 2000). The quality management improvements mentioned in this chapter will need to be adopted to have quality management institutionalised at SasTech.

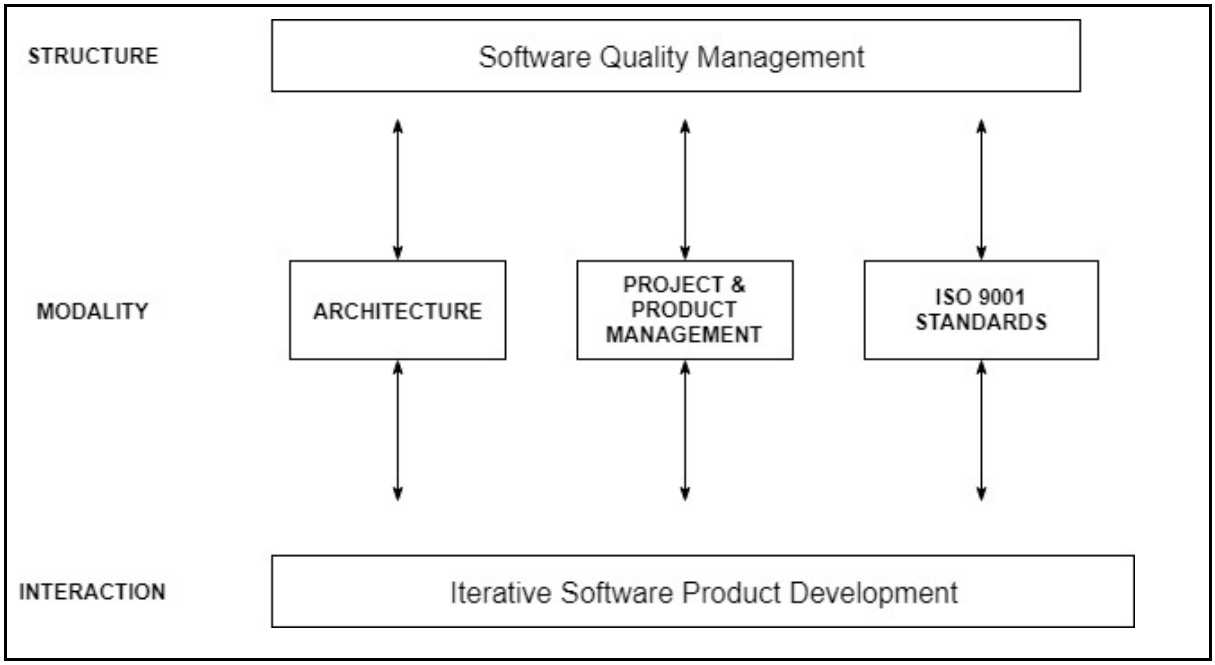


Figure 5.2: General framework: software quality management -in- practice

As illustrated by figure 5.2 above, the architecture and management influence contribute to how processes are defined. Technology structure determines how the process should be and what quality aspects you need to measure. This enables:

> *Setting performance indicators (KPIs) by setting targets and then measure it to give you expected results.* (LT- senior developer)

The general quality management rule is:

> *To get requirements specified as clear and precise as possible, get prioritised and into development and testing to make sure it's a working deliverable then release to customers.* (TO- product manager)

The effectiveness of quality management systems is determined by processes established. If the quality management is institutionalised in an organisation, this may lead to a quality software product. Upskilling of personnel, new technologies, time management, management support, effective communication lines and best practices combined with ISO 9001 requirements lead to an adequate quality management.

However, this is not stable, new processes will need to be enacted over time. The structurational framework helps with learning, influence and dependence across organisations. As perceived by DoT to understand and interpret an effective institutionalisation of quality management process in iterative software development, the outcome of this study indicates that institutionalisation of an effective quality management process has many parts and is achieved through the production of quality management strategies.

5.6 Summary

The institutionalisation of quality management in iterative software development entails quality management practices in an organisation by determining the factors that influence the quality management process using duality of technology model derived from ST by Giddens. All the results were analysed through the lens of the three modalities facilities, norms and interpretive schemes.

This chapter outlined the results and interpretation of the findings generated from the qualitative approach. 22 interviews were carried out. The data was analysed using thematic analysis. Based on the underpinning concept of the duality of technology model, key

components of the theory of structuration were used as a theoretical lens through which the research questions were addressed and answered.

Quality management as a structure enacts best practices and facilities which through routine or daily use and interaction with the iterative development process will structure the quality management process. Thus, the quality management process through its use will produce and reproduce a new process enacting quality management in practice (QMIP).

CHAPTER SIX

CONCLUSION AND RECOMMENDATION

6.1 Introduction

The aim was to explore the factors influencing the inadequacy of quality management in the software development process. The research was based on a case of a software development organisation in Cape Town, South Africa. 22 interviews were conducted at this organisation called SasTech and participants comprised of product managers, testers and developers and UX designers.

Thematic analysis was then used to interpret the findings and duality of technology model was used as a theory of analysis to interpret the findings. Several themes were created which included the factors which influence the quality management process, challenges and improvements. A general framework was then proposed in chapter five to institutionalise quality management in iterative software product development at SasTech.

This chapter provides a conclusion of the research study and provides recommendations.
To conclude the research report, this chapter is organised as follows:

(a) Overview of the research
(b) Research questions revisited
(c) Research contributions
(d) Conclusion
(e) Recommendation
(f) Limitation
(g) Further research
(h) Summary

6.2 Overview of the research

The research has six chapters summarised in figure 6.1 below:

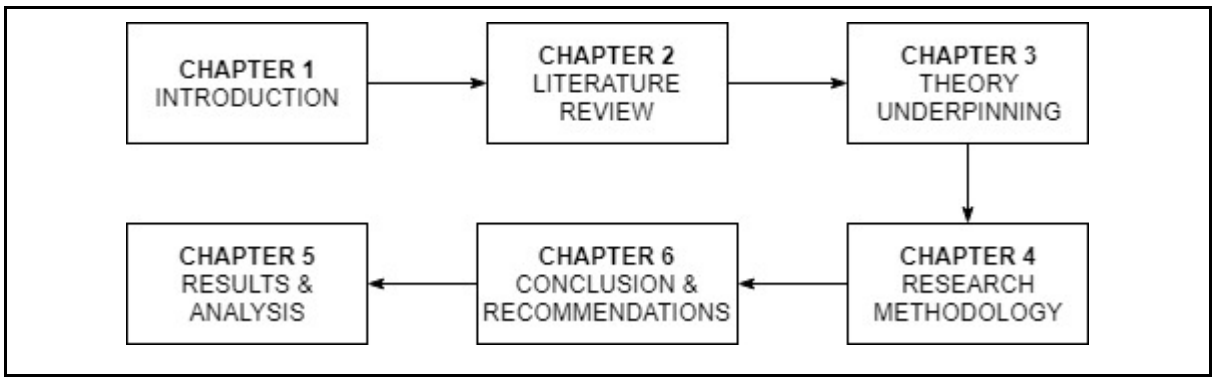


Figure 6.1: Thesis summary

Chapter One as the introductory chapter outlined the structure of the research study conducted and research context. The aim of the study was to introduce the importance of quality software product delivery and its benefits to both the customer and the organisation developing the product. This chapter explained how rapid technology is changing and how complex software products are becoming. With this development, the pressure is put on organisations to innovate while making sure that there are effective software product development methodologies adopted such as ISD with a working quality management system.

Therefore, the aim of this study was to identify some of the quality management challenges that are faced by product development teams of the selected organisation to determine what factors influence the quality management process of software product development. The organisation under study uses an iterative process to deliver its products. The intent of this study was to produce a guideline to an approach that can mitigate some of these challenges faced by the selected organisation as well as other similar organisations in the industry. Following the research aim, was the research problem, theoretical underpinning, literature review and research methodology overviews, ethical considerations, contributions and limitations.

Chapter Two addressed literature related to the research problem. It started with a general overview of common challenges, software development process and how its evolving, XP and Scrum methods, software quality management and its components and software delivery standard - ISO 9001. Related studies were also discussed bringing out the gap which this study addressed.

The purpose of chapter three was to discuss the theoretical underpinning of this study. How quality management in software development can be analysed using theoretical lens. Structuration theory was introduced and all its related elements in, structure, agency and duality of structure. Duality of technology model was discussed in detail as this was the model used to analyse the results of this study. The importance of it in understanding how meanings

are driven by the structure and how structure influence actions and vice versa. The concepts explained in this chapter influenced the research methodology in chapter four and drawing meanings in chapter five.

Chapter Four discussed the research methodology used in the study. A qualitative case study approach was used. The research process, approach, philosophy, methodology, research and sampling design were outlined. To collect data, a survey was done, and 22 interviews were conducted with ethical considerations. Data was recorded, and transcribed.

Chapter Five outlined the results and interpretation of the findings generated from the qualitative approach. The data was analysed using thematic analysis. Based on the underpinning concept of the duality of technology model, key components of the theory of structuration were used as a theoretical lens through which the research questions were addressed and answered. A general framework for quality management in ISD was proposed.

Chapter Six provides a conclusion of the research study and recommendations. To conclude the research report, this chapter is further organised as follows: the overview of the research, research questions revisited, research contributions, conclusion, recommendation, limitation, further research and summary.

Orlikowski (2000) acknowledges that the institutionalisation of information systems is not stable and through its use new structures are enacted. Thus, in this case, quality management is constituted of many enactments. Interpretive schemes which reflect the architecture, facilities which reflect technologies and human resources and norms which are the best practices.

6.3 Research questions revisited

Investigating the research problem of inadequate quality management in ISD. The research questions are outlined below:

6.3.1 Research Question 1: How is quality management in software product development influenced?

Factors influencing software quality management were identified as documentation of correct requirements, design, planning, development process and technology, testing release process, management support, maintenance support and upgrades, usability and user experience, timelines and process ownership.

6.3.2 Research Question 2: How are socio-technical processes embedded in the quality management process of software product development?

Socio-technical processes are embedded in software quality management through requirements and feedback gathering which ensure the product of the right quality is produced with each iteration.

The use of the right tools, for example, Selenium for automation testing, new languages such as Scala and Go and architecture that allows growth and innovation, this will improve software quality management. However, this is only possible when there are enough people within the organisation with the right skills and experience. Also, with continuous learning.

Duality of technology model concludes the importance of merging social and technology systems to have a working solution that is of the right quality. Socio-technical systems are guided rules and policies (norms) and lines of communication (interpretive schemes) of the organisation.

6.3.3 Research Question 3: Why are software development processes ineffective?

Drawing from the literature and results of the study, lack of proper quality management system was identified as the reason why software development process was ineffective. If the organisation doesn't have a proper process or doesn't ensure that the process is followed, they are likely to produce a substandard product. Challenges identified include poor planning, poor design, limited resources, tight deadlines and lack of documentation causing a lot of problems for the organisation under study which leads to loss of opportunities to competing organisations.

6.3.4 Research Question 4: How can quality management in software product development be guided to meet customer expectations?

Through certain practices, the right architecture and guidelines of ISO 9001 a working software quality management system can be developed. The best practices discussed and recommended are planning, documentation, collaboration, setting KPIs, automated testing and through testing, the right technologies and involvement of the customer or user of the product.

In section 5.6 a general framework was illustrated (figure 5.2) and discussed that SasTech can use in designing their quality management system. Having a working software quality management system will ensure that the organisation produce quality products that meet customers' needs.

Referencing answers given to research question, contributions made by this study are discussed in the following section.

6.4 Research Contributions

This study has contributed to the following areas: theoretical, methodological, practical and which are discussed in the sections below.

6.4.1 Theoretical Contributions

Theoretical contribution is about how with the theory underpinning, this study accomplished its outcome and implication. The research aim was to identify factors that influence software quality management in iterative software development. Using DoT model of structuration theory, a general framework was created to guide literature review, data collection and data analysis. The use of DoT model as the underpinning theory to derive a general framework to identify factors that influence software quality management is the theoretical contribution of this study.

6.4.2 Methodological Contributions

The empirical nature of this research required interpretive inquiry as data analysis required both researcher's own interpretations as well as participants'. Since the study was based on an individual organisation, SasTech a software development organisation, an intensive investigation and analysis was required which made a case study approach most ideal.

A qualitative approach was employed in this study because the subject materials needed to be evaluated in greater detail and research data needed to come from participants' experience and knowledge. The data gathered had to be of quality to draw meanings from it. Interviews were the choice for data collection to gain deep insights into the research problem of inadequate quality management in ISD. Thematic analysis was used to analyse the data generated from the qualitative approach.

The methodological contributions in this study are the approach of taking an interpretive stance, a qualitative case study approach with the use of semi-structured interviews and thematic analysis to address the research problem to serve as a guide for other researchers.

6.4.3 Practical Contributions

The aim was to explore the factors influencing the inadequacy of quality management in software development process a case of a software development organisation in Cape Town, South Africa. To address the aim, the main objective was:

> "*to investigate the factors influencing quality management process on the development of quality software products*."

The practical contribution of this study is the institutionalisation of quality management in ISD. The general framework proposed will assist software development organisations involved in iterative development to create an effective and efficient software quality management system to ensure the delivery of quality software products.

6.5 Recommendations

Emerging from the results, the following is recommended to assist SasTech in improving their software product development process:

(a) Creating culture where processes are documented clearly.
(b) Requirements that are well articulated and documented in central repository.
(c) Continuous education for all employees involved.
(d) Improve communication and collaboration as these are necessary at all steps of the software development life cycle through having more review meetings to ensure nothing gets missed.
(e) Better prioritisation- critical bugs; work tasks; backlog grooming, backlog with well-defined user stories.
(f) Start over- redo legacy systems where possible.
(g) Moratorium/code freeze after development is complete before release
(h) Making time for quality management planning
(i) Communicating on workload and timelines in advance
(j) Stakeholder consensus.
(k) Unrushed implementation or development.
(l) Ensuring optimal error handling/trapping in the source code
(m) Adopt Spotify engineering culture- creating autonomous teams (see appendix E)
(n) Unit testing.
(o) Peer reviews for efficiency and best practices for example, pair programming.
(p) Code reviews.
(q) Fully test plan with pre-defined outcomes and well-defined edge cases.
(r) Test script/ procedure outlining how to conduct tests.
(s) Making a checklist of all use cases and taking time of thinking of the edge cases.

(t) Technical testing - developers must test code before handing over to QA.
(u) UAT before release.
(v) Pilot testing.
(w) Beta release.
(x) Automated testing.
(y) Risk-based testing
(z) Load & stress testing

Planning, collaboration, testing techniques, automation, technologies, customer involvement and frequent communication within teams were, key improvements which were common in the transcripts.

6.6 Limitations and further research

This research was a case study of one organisation in Cape Town, South Africa. Therefore, findings may not be generalised for all organisations, but this will provide a general framework which will consequently make a valuable contribution. The intent is to reduce the size of the sample but to provide a much-detailed analysis to understand the phenomenon and then conceptualising the findings of which future studies can be built on. The main factors driving the decision to limit the size of the study area was the available time and resources as the study has to be completed within two years under a limited budget.

Since this is one case study, for further research the general framework can be tested with other organisations and in other regions to test its applicability.

6.7 Summary

Quality management is not institutionalised at SasTech. Using duality of technology model as a model of analysis, a general framework for software quality management at SasTech was proposed to improve the software delivery process. Through facilities (technologies and human resources), norms (best practices) and interpretive schemes (architecture) software quality management can be institutionalised.

REFERENCES

Abrahamsson, P., Conboy, K. & Wang, X. 2009. “Lots done, more to do”: the current state of agile systems development research. *European Journal of Information Systems*, 18(4), pp.281–284.

Abrahamsson, P., Salo, O., Ronkainen, J. & Warsta, J., 2017. Agile software development methods: Review and analysis. *arXiv preprint arXiv:1709.08439*.

Alahyari, H., Svensson, R. B. & Gorschek, T. 2016. A study of value in agile software development organisations. *The Journal of Systems & Software*, 125:271–288. doi: 10.1016/j.jss.2016.12.007.

Albano, R., Masino, G. & Maggi, B. 2010. The relevance of Giddens' structuration theory for organisational research.

Alhassan, A., Alzahrani, W. & Abdulaziz, A. 2017. Total Quality Management for Software Development. *International Journal of Computer Applications*, 158(5):975–8887.

Allison, I. & Merali, Y. 2007. Software process improvement as emergent change: A structurational analysis. *Information and software technology*, *49*(6), pp.668-681.

Alshammri, M. 2013. Problems in Software Quality Assurance and Reasons. *IJCSI International Journal of Computer Science Issues*, *10*(1), pp.325-327.

Anyan, F. 2013. The influence of power shifts in data collection and analysis stages: A focus on qualitative research interview. *The Qualitative Report*, *18*(18), p.1.

Aregbesola, K., Akinkunmi, B.O. & Akinola, O.S. 2011. Process maturity assessment of the Nigerian Software Industry. *Int J Adv Eng Technol*, *1*, pp.10-25.

Aronson, J. 1995. A pragmatic view of thematic analysis. *The qualitative report*, *2*(1), pp.1-3.

Baig, J.J.A., Shah, A. & Sajjad, F. 2017. Evaluation of agile methods for quality assurance and quality control in ERP implementation. In *Intelligent Computing and Information Systems (ICICIS), 2017 Eighth International Conference on* (pp. 252-257). IEEE.

Barata, J. & da Cunha, P. R. 2015. Do you walk the talk in quality culture? *Journal of Information and Organisational Sciences*, 39(2):137–150.
Barreiro, P.L. & Albandoz, J.P., 2001. Population and sample. Sampling techniques. *Management mathematics for European schools*, *6*.

Baškarada, S. 2014. Qualitative Case Study Guidelines. *Qualitative Report,* 19(40): 1- 25.

Baum, T., Liskin, O., Niklas, K. & Schneider, K. 2016. Factors influencing code review processes in industry. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering* (pp. 85-96). ACM.

Baxter, P. & Jack, S., 2008. Qualitative case study methodology: Study design and implementation for novice researchers. *The qualitative report*, *13*(4), pp.544-559.

Baxter, G. & Sommerville, I. 2011. Socio-technical systems: From design methods to systems engineering. *Interacting with Computers,* 23: 4-17.

Bertolotti, I.C. & Hu, T. 2016. *Embedded Software Development: The Open-Source Approach*. CRC Press.

Bourque, P. & Fairley, R.E. 2014. *Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0*. IEEE Computer Society Press.

Boyatzis, R.E. 1998. *Transforming qualitative information: Thematic analysis and code development*. sage.

Broger, D., 2011. *Structuration theory and organization research*. Na.

Bryant, C.G. 1999. The uses of Giddens' structuration theory.

Camisón, C. & Puig-Denia, A. 2016. Are quality management practices enough to improve process innovation? *International Journal of Production Research*, *54*(10), pp.2875-2894.

Carlsson, S.A. 2006. Towards an information systems design research framework: A critical realist perspective. In *International Conference on Design Science Research in Information Systems and Technology (DESRIST)*(pp. 192-212).

Chevers, D. A. & Grant, G. 2017. Developers Views on Information Systems Quality and Success in Canadian Software Development. *JISTEM - Journal of Information Systems and Technology Management,* TECSI Laboratório de Tecnologia e Sistemas de Informação - FEA/USP, 14(1):3–20. doi: 10.4301/s1807-17752017000100001.

Claps, G.G., Svensson, R.B. & Aurum, A. 2015. On the journey to continuous deployment: Technical and social challenges along the way. *Information and Software technology*, *57*, pp.21-31.

Clutterbuck, P., Rowlands, T. & Seamons, O. 2008. A case Study of SME web application development via agile methods. *2nd European Conference on Information Management and Evaluation, ECIME 2008*, 12(1):77–88.

Coallier, F. 1994. How ISO 9001 fits into the software world. *IEEE Software*, *11*(1), pp.98-100.

Cohen, B. & College, L. 2014. Introducing Engineering as a Socio-technical Process Introducing Engineering as a Socio-technical Process.

Colomo-Palacios, R., Soto-Acosta, P., Mishra, A. & García-Crespo, Á. 2011. Software quality management improvement through mentoring: an exploratory study from GSD projects. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*(pp. 190-199). Springer, Berlin, Heidelberg.

Creswell, J.W. & Creswell, J.D. 2017. *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications.

DeSanctis, G. & Poole, M.S. 1994. Capturing the complexity in advanced technology use: Adaptive structuration theory. *Organization science*, *5*(2), pp.121-147.

De Villiers, M.R. 2012. Models for Interpretive Information Systems Research, Part 2: Design Research, Development. *Research Methodologies, Innovations and Philosophies in Software Systems Engineering and Information Systems*, p.238.

Dingsøyr, T. & Lassenius, C. 2016. Emerging themes in agile software development: Introduction to the special section on continuous value delivery. *Information and Software Technology*, *77*, pp.56-60.

Doshi, V.P. & Patil, V. 2016. Competitor driven development: Hybrid of extreme programming and feature driven reuse development. In *Emerging Trends in Engineering, Technology and Science (ICETETS), International Conference on* (pp. 1-6). IEEE.

Dovleac, R. and Ionică, A., 2017. Quality Management techniques embedded in Agile Project Development. In MATEC Web of Conferences (Vol. 121, p. 05003). EDP Sciences.

Draper, A. 2004. The principles and application of qualitative research. *Proceedings of the Nutrition Society*, 63(4): 641-646. doi:10.1079/PNS2004397.

Drew, C.J., Hardman, M.L. & Hosp, J.L. 2007. *Designing and conducting research in education*. Sage Publications.

Ebert, C., Abrahamsson, P. & Oza, N. 2012. Lean software development. *IEEE Software*, *29*(5), pp.22-25.

Englund, H., Gerdin, J. & Burns, J. 2011. 25 years of Giddens in accounting research: achievements, limitations and the future. *Accounting, Organizations and Society*, *36*(8), pp.494-513.

Femmer, H., Fernández, D.M., Wagner, S. & Eder, S. 2017. Rapid quality assurance with requirements smells. *Journal of Systems and Software*, *123*, pp.190-213.

Fitzgerald, B. & Stol, K.J. 2015. Continuous software engineering: A roadmap and agenda. *The Journal of Systems & Software*, 123:176–189. doi: 10.1016/j.jss.2015.06.063.

Fitzpatrick, R., 1996. Software quality: definitions and strategic issues.

Fitzpatrick, R., Smith, P., & O'Shea, B. 2004. Software Quality Challenges, *Proceedings of the Second Workshop on Software Quality* at the 26th International Conference on Software Engineering (ICSE 2004), Edinburgh, Scotland. Stevenage, Herts, UK: IEE.

Flora, H.K. & Chande, S.V. 2014. A systematic study on agile software development methodologies and practices. *International Journal of Computer Science and Information Technologies*, *5*(3), pp.3626-3637.

Fowler, F.J. 2002. *Survey research methods. Thousand Oaks*, CA: Sage Publications.

Galetto, M., Franceschini, F. & Mastrogiacomo, L., 2017. ISO 9001 certification and corporate performance of Italian companies. *International Journal of Quality & Reliability Management*, *34*(2), pp.231-250.

Gehman, J. 2008. Structuration theory summary. *Retrieved April 1*, p.2016.

Gerring, J. 2004. What is a case study and what is it good for? *American political science review*, *98*(2), pp.341-354.

Giardino, C., Paternoster, N., Unterkalmsteiner, M., Gorschek, T. & Abrahamsson, P. 2016. Software development in startup companies: the greenfield startup model. *IEEE Transactions on Software Engineering*, *42*(6), pp.585-604.

Giddens, A. 1984. *The Constitution of Society: Outline of the Theory of Saturation*. University of California Press.

Gill, N. S. 2005. Factors affecting effective software quality management revisited. *ACM SIGSOFT Software Engineering Notes*, 30 (2).

Gill, P., Stewart, K., Treasure, E. & Chadwick, B. 2008. Methods of data collection in qualitative research: interviews and focus groups. *British dental journal*, *204*(6), p.291.

Gray, D.E. 2013. *Doing research in the real world*. Sage.

Hamman, M. 2009. Organizational Agility Capability, Structuration Theory, and Social Practices.

Hancock, B., Windridge, K. & Ockleford, E. 2009. An introduction to qualitative research. The National Institute of Health Research. *Research Development Society for East Midlands Yorkshire and Humber. University of Nottingham, Nottingham*.

Harrington, A., 2005. *Modern social theory*. Oxford: Oxford University Press.

Hashmi, S. I. & Baik, J. 2007. Software quality assurance in XP and spiral - A comparative study. *Proceedings - The 2007 International Conference on Computational Science and its Applications,* ICCSA 2007, 367–374. doi: 10.1109/ICCSA.2007.69.

Highsmith, J.A. & Highsmith, J. 2002. *Agile software development ecosystems* (Vol. 13). Addison-Wesley Professional.

Hinchey, M.G. & Sterritt, R. 2006. Self-managing software. Computer, 39(2), pp.107-109.

Hove, S.E. & Anda, B., 2005. Experiences from conducting semi-structured interviews in empirical software engineering research. In *Software metrics, 2005. 11th ieee international symposium* (pp. 10-pp). IEEE.

Huo, M., Verner,J., Zhu, L. & Ali Babar, M. 2004. Software Quality and Agile Methods. *Proceedings of the 28th Annual International Computer Software and Applications Conference*, COMPSAC 2004, 520–525.

Isaacs, A.N. 2014. An overview of qualitative research methodology for public health researchers. *International Journal of Medicine and Public Health*, *4*(4).

Iyamu, T. 2017. Understanding the Complexities of Enterprise Architecture through Structuration Theory. *Journal of Computer Information Systems*, pp.1-9.

Jalote, P. & Agrawal, N. 2005. Using defect analysis feedback for improving quality and productivity in iterative software development. In *Information and Communications Technology, 2005. Enabling Technologies for the New Knowledge Society: ITI 3rd International Conference on* (pp. 703-713). Ieee.

Jamsutkar, K., Patil, V. & Chawan, P. M. 2012. Software Project Quality Management, 2(3):686–690.

Janes, A., Lenarduzzi, V. & Stan, A.C. 2017. A continuous software quality monitoring approach for small and medium enterprises. In *Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering Companion* (pp. 97-100). ACM.

Jinzenji, K., Hoshino, T., Williams, L. & Takahashi, K. 2013. An experience report for software quality evaluation in highly iterative development methodology using traditional metrics. 2013 IEEE 24th International Symposium on Software Reliability Engineering (ISSRE).
Johannesson, P. & Perjons, E. 2014. A Method Framework for Design Science Research. In *an Introduction to Design Science,* 75-89. Springer International Publishing.

Jones, M.R. & Karsten, H. 2008. Giddens's structuration theory and information systems research. *MIS quarterly*, *32*(1):127-157.

Jovanovic, V. & Shoemaker, D. 1997. ISO 9001 standard and software quality improvement. *Benchmarking for Quality Management & Technology*, *4*(2), pp.148-159.

Kadam, P. & Bhalerao, S. 2010. Sample size calculation. *International journal of Ayurveda research*, 1(1):55.

Karambir & Sharma, A. 2016. A review of agile methodology in software development. *International Research Journal of Engineering and Technology*, 3(3):1325–1329.
Karout, R. & Awasthi, A., 2017. Improving software quality using Six Sigma DMAIC-based approach: a case study. *Business Process Management Journal*, *23*(4), pp.842-856.

Kassab, M., DeFranco, J.F. & Laplante, P.A. 2017. Software Testing: The State of the Practice. *IEEE Software*, *34*(5), pp.46-52.

Kaur, R. & Sengupta, J. 2013. Software process models and analysis on failure of software development projects. *arXiv preprint arXiv:1306.1068*.

Khan, A., Keung, J., Niazi, M., Hussain, S. & Ahmad, A. 2017. Systematic literature review and empirical investigation of barriers to process improvement in global software development: Client–vendor perspective. *Information and Software Technology*, 87:180–205. doi: 10.1016/j.infsof.2017.03.006.

Khalane, T. & Tanner, M. 2013. Software quality assurance in Scrum: The need for concrete guidance on SQA strategies in meeting user expectations. In *Adaptive Science and Technology (ICAST), 2013 International Conference,* 1-6. IEEE.

Knippers, D. 2011. Agile Software Development and Maintainability. In *15th Twente Student Conf*.

Kothari, C.R. 2004. *Research methodology: Methods and techniques*. New Age International.

Kruchten, P. 2008. On Software Architecture, Agile Development, Value and Cost Story of a failure. *Engineering*, (May):1–14.

Kuhrmann, M., Diebold, P., Münch, J., Tell, P., Garousi, V., Felderer, M., Trektere, K., McCaffery, F., Linssen, O., Hanser, E. & Prause, C.R. 2017. Hybrid software and system development in practice: waterfall, scrum, and beyond. In *Proceedings of the 2017 International Conference on Software and System Process* (pp. 30-39). ACM.

Lee, S.N., Ko, D., Park, S. & Kim, S. 2014. An approach to building domain architectures using domain component model and architectural tactics. *International Journal of Engineering Systems Modelling and Simulation*, *6*(1-2), pp.54-61.

Lee, H.L. & Chin, K.S. 2017. Development of a Systematic Model for Improving Software Quality in the Financial Service Organization. In *Management Challenges in a Network Economy: Proceedings of the MakeLearn and TIIM International Conference 2017* (pp. 361-370). ToKnowPress.

Lewis, S. 2015. Qualitative inquiry and research design: Choosing among five approaches. *Health promotion practice*, *16*(4), pp.473-475.

Lewis, W.E. 2016. *Software testing and continuous quality improvement.* CRC press.

Lindstrom, L. & Jeffries, R. 2004. Extreme programming and agile software development methodologies. Information systems management, 21(3), pp.41-52.

Loureiro-Koechlin, C. 2008. A theoretical framework for a structuration model of social issues in software development in information systems. *Systems Research and Behavioral Science*, *25*(1), pp.99-109.

Mafuwane, B.M. 2012. *The contribution of instructional leadership to learner performance* (Doctoral dissertation, University of Pretoria).

Mahdavi, G. & Daryaei, A.A. 2015. Giddens' structuration theory, corporate governance and audit marketing. *Ludus Vitalis*, *11*(1), pp.1-7.

Mäkinen, S., Leppänen, M., Kilamo, T., Mattila, A.L., Laukkanen, E., Pagels, M. and Männistö, T. 2016. Improving the delivery cycle: A multiple-case study of the toolchains in Finnish software intensive enterprises. *Information and Software Technology*, *80*, pp.175-194.

Manikas, K. & Hansen, K. M. 2013. The Journal of Systems and Software Software ecosystems – A systematic literature review. *The Journal of Systems and Software*, 86:1294–1306. doi: 10.1016/j.jss.2012.12.026.

Mansoor, S., Bhutto, A., Bhatti, N., aamir Patoli, N. & Ahmed, M. 2017. Improvement of students' abilities for quality of software through personal software process. In *Innovations in Electrical Engineering and Computational Technologies (ICIEECT), 2017 International Conference on* (pp. 1-4). IEEE.

Marques, R., Costa, G., Silva, M. & Gonçalves, P., 2017. A survey of failures in the Software Development Process.

Mateen, A., Jahanzaib, M. & Iqbal, N. 2017. The Role of Quality Assurance in Software Development Projects: Project Failures and Business Performance. *International Journal of Management, IT & Engineering Journal Homepage*, 7(2): 2249–558.

McFall, D., Wilkie, F.G., McCaffery, F., Lester, N. & Sterrit, R. 2017. Software Processes and Process Improvement in Northern Ireland.

McIntosh, S., Kamei, Y., Adams, B. & Hassan, A.E. 2014. The impact of code review coverage and code review participation on software quality: A case study of the qt, vtk, and itk projects. In *Proceedings of the 11th Working Conference on Mining Software Repositories* (pp. 192-201). ACM.

McMichael, B. & Lombardi, M. 2007. ISO 9001 and Agile development. In *Agile Conference (AGILE), 2007* (pp. 262-265). IEEE.

Mendez, E.L. & Pinto-Roa, D.P. 2018. Optimization model for assigning tasks in Scrum Agile Development (work in progress). *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, *6*(1).

Mkansi, M. & Acheampong, E.A. 2012. Research philosophy debates and classifications: students' dilemma. *Electronic journal of business research methods*, *10*(2), pp.132-140.

Mohammed, N., Munassar, A. & Govardhan, A. 2010. A Comparison Between Five Models of Software Engineering. *International Journal of Computer Science Issues*, 7(5).

Moniruzzaman, A.B.M. & Hossain, D.S.A. 2013. Comparative Study on Agile software development methodologies. *arXiv preprint arXiv:1307.3356*.

Mora, M., Gelman, O., Steenkamp, A. & Raisinghani, M.S. 2012. Research Methodologies, Innovations and Philosophies in Software Systems Engineering and Information Systems, *IS*

*Research. Action Research, Grounded Theory- A Meta-Study and Examples :*222–237. doi: 10.4018/978-1-4666-0179-6.ch011.

Munyaradzi, Z., Maxmillan, G. & Amanda, M.N. 2013. Effects of Web Page Contents on Load Time over the Internet. *Int Journal of Science and Research*, *2*(9), pp.75-79.

Murgia, A., Tourani, P., Adams, B. & Ortu, M. 2014. Do developers feel emotions? an exploratory analysis of emotions in software artifacts. In *Proceedings of the 11th working conference on mining software repositories* (pp. 262-271). ACM.

Myeko, Z. 2014. *A systematic approach for managing design changes on global collaborative projects: a case study analysis of the Medupi structural steel* (Doctoral dissertation).

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

Naidoo, L. 2009. A Structuration Theory Analysis of the Refugee Action Support Program in Greater Western Sydney. *Australian Journal of Teacher Education*, 34(4):40–50. doi: 10.14221/ajte.2009v34n4.5.

Nanda, V. 2016. *Quality management system handbook for product development companies*. CRC Press.

Nicholls, D. 2017. Qualitative research. Part 3: Methods. International Journal of Therapy and Rehabilitation, 24(3), pp.114-121.

Norman, M. and Venter, I., 2016, September. Factors for developing a software industry. In *Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists* (p. 29). ACM.

Oberscheven, F.M. 2013. Software Quality Assessment in an Agile Environment. *Faculty of Science of Radboud University in Nijmegen.*

O'Leary, C. & Gordon, D. 2009. Universal design, education and technology.

Oppong, S. 2014. Between Bandura and Giddens: Structuration Theory in Social Psychological Research? *Psychological Thought*, *7*(2), pp.111-123.

Orlikowski, W.J. & Robey, D. 1991. Information technology and the structuring of organizations. *Information systems research*, *2*(2), pp.143-169.

Orlikowski, W.J. 1992. The duality of technology: Rethinking the concept of technology in organizations. *Organization science*, *3*(3), pp.398-427.

Orlikowski, W.J. 2008. Using technology and constituting structures: A practice lens for studying technology in organizations. In *Resources, co-evolution and artifacts* (pp. 255-305). Springer, London.

Osmundson, J.S., Michael, J.B., Machniak, M.J. & Grossman, M.A. 2003. Quality management metrics for software development. *Information & Management*, *40*(8), pp.799-812.

Palinkas, L.A., Horwitz, S.M., Green, C.A., Wisdom, J.P., Duan, N. & Hoagwood, K. 2015. Purposeful sampling for qualitative data collection and analysis in mixed method implementation research. *Administration and Policy in Mental Health and Mental Health Services Research*, *42*(5):533-544.

Pandey, A.K., Saxena, A.S. & Pandey, R. 2013. Framework of Software Quality Management Using Object Oriented Software Agent.

Patwardhan, A. 2016. Structured unit testable templated code for efficient code review process. *arXiv preprint arXiv:1610.05726*.

Popović, T. 2015. Getting ISO 9001 certified for software development using scrum and open source tools: a case study. *Tehnički vjesnik*, *22*(6), pp.1633-1640.

Poole M. S., DeSanctis G. 2004. Structuration theory in information systems research: Methods and controversies. In Whitman M. E., Woszczynski A. B. (Eds.), *Handbook of information systems research*, 206-249. Hershey, PA: Idea Group.

Pressman, R.S. 2010. Software Engineering: A Practitioner's Approach, 7/e, RS Pressman & Associates. *Inc., McGraw-Hill, ISBN*, *73375977*.

Qasaimeh, M. & Abran, A. 2013. An audit model for ISO 9001 traceability requirements in agile-XP environments. *Journal of Software*, *8*(7), pp.1556-1567.

Quadri, A.T. Abubakar, M. & Sirshar, M. 2015. Software Quality Assurance in Component Based Software Development–a Survey Analysis. *International Journal of Computer and Communication System Engineering*, *2*(2), pp.305-315.

Rabionet, S.E. 2011. How I learned to design and conduct semi-structured interviews: An ongoing and continuous journey. *The Qualitative Report*, *16*(2), p.563.
Rajasekar, S., Philominathan, P. & Chinnathambi, V., 2006. Research methodology. *arXiv preprint physics/0601009*.

Rajkuma, G. & Alagarsamy, D.K. 2013. The most common factors for the failure of software development project. *The International Journal of Computer Science & Applications (TIJCSA) Volume, 1.*

Ramakrishnan, A. & Manjula, R. 2016. Perceptual Difference for Safer Continuous Delivery.

Rizvi, Q. M. 2014. Software Quality Assurance: An Introductory Note. Software Quality Assurance, (March). *https://doi.org/10.1016/B978-0-12-802301-3.00012-0.*

Rose, J. 1998. Evaluating the contribution of structuration theory to the information systems discipline. In *6th European Conference on Information Systems* (Vol. 2, pp. 910-924).

Rose, J. & Scheepers, R. 2001. Structuration theory and information system development - frameworks for practice. *ECIS Proceedings*, Bled.

Rowley, J. 2002. Using case studies in research. *Management Research News*, 25(1): 16-27.

Royce, W. Bittner, K. & Perrow, M. 2009. *The economics of iterative software development: Steering toward better business results*. Pearson Education.

Saunders, M.N.K. & Tosey, P.C. 2013. The layers of research design. *Rapport*, (Winter), pp.58-59.

Scarpino, J. & Kovacs, P. 2008. An Analysis of a Software Quality Assurance Tool's Implementation: A Case Study. *Journal of the International Association for Computer Information Systems*, *9*(2), p.9.

Schön, E.M., Thomaschewski, J. & Escalona, M. J. 2017. Agile Requirements Engineering: A systematic literature review. *Computer Standards & Interfaces*, 49:79–91. doi: 10.1016/j.csi.2016.08.011.

Schüler, K., Trogus, R., Feilkas, M. & Kinnen, T. 2015. Managing product quality in complex software development projects. In *Proceedings of the Embedded World Conference*.

Sedano, T., Ralph, P. & Péraire, C. 2017. Software development waste. In *Proceedings of the 39th International Conference on Software Engineering* (pp. 130-140). IEEE Press.

Shmueli, O. & Ronen, B. 2017. Excessive software development: Practices and penalties. *International Journal of Project Management*, *35*(1), pp.13-27.

Shrivastava, S.V. 2010. Distributed agile software development: A review. *arXiv preprint arXiv:1006.1955*.

Silva, F.S., Soares, F.S.F., Peres, A.L., de Azevedo, I.M., Vasconcelos, A.P.L., Kamei, F.K. & de Lemos Meira, S.R., 2015. Using CMMI together with agile software development: A systematic review. *Information and Software Technology*, *58*, pp.20-43.

Silva, F. A., & Kazmierkowski, M. 2016. Book News, (march), 57–58.

Sommerville, I. 2010. *Software engineering*. New York: Addison-Wesley.

Sommerville, I. 2011. *Software Engineering Ninth Edition*. Pearson Education Inc. publishing as Addison-Wesley, Boston: Massachusetts 02116.

Soundararajan, S., Arthur, J.D. and Balci, O., 2012, August. A methodology for assessing agile software development methods. In *Agile Conference (AGILE), 2012* (pp. 51-54). IEEE.

Sowunmi, O.Y. & Misra, S. 2015. An empirical evaluation of software quality assurance practices and challenges in a developing country. In *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on* (pp. 867-871). IEEE.

Sowunmi, O.Y., Misra, S., Fernandez-Sanz, L., Crawford, B. & Soto, R. 2016. An empirical evaluation of software quality assurance practices and challenges in a developing country: a comparison of Nigeria and Turkey. *SpringerPlus*, *5*(1), p.1921.

Stålhane, T. & Hanssen, G.K. 2008. The application of ISO 9001 to agile software development. In *International Conference on Product Focused Software Process Improvement* (pp. 371-385). Springer, Berlin, Heidelberg.

Standish Group (2015) *Chaos Report.* Available at: https://www.infoq.com/articles/standish-chaos-2015 (Accessed: 28 October 2015).

Stark, J. 2015. Product Lifecycle Management. 21st Century Paradigm for Product Realisation. *Springer International Publishing, AG Switzerland,* 1:17440-2.

Starman, A. B. 2013. The case study as a type of qualitative research. *Journal of Contemporary Educational Studies*, (1):28–43.

Stelzer, D., Mellis, W. & Herzwurm, G. 1997. A critical look at ISO 9000 for software quality management. *Software Quality Journal*, *6*(2), pp.65-79.

Stillman, L.J.H. 2006. *Understandings of technology in community-based organisations: A structurational analysis* (Doctoral dissertation, Monash University).

Stones, R. 2007. Structure and agency. In G. Ritzer (Ed.). *Blackwell Encyclopedia of Sociology* :1-4. doi: doi:10.1111/b.9781405124331.2007.

Sutton, J. & Austin, Z. 2015. Qualitative research: data collection, analysis, and management. *The Canadian journal of hospital pharmacy*, *68*(3), p.226.

Tan, T., Li, Q., Boehm, B., Yang, Y., He, M. & Moazeni, R. 2009. Productivity trends in incremental and iterative software development. In *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement* (pp. 1-10). IEEE Computer Society.

Theron, P.M. 2015. Coding and data analysis during qualitative empirical research in Practical Theology. *In die Skriflig*, 49(3):1-9.

Thomas, P. 2010. Research Methodology and Design, Research Methodology and Design*,* 291–334. doi: 10.1515/9783110215519.82.

Thompson, S.K. 2012. Stratified sampling. *Sampling, Third Edition*, pp.139-156.'

Timbrell, G.T., Delaney, P., Chan, T., Yue, W.A. & Gable, G. 2005. A structurationist review of knowledge management theories. Association for Information Systems.

Tomar, A.B. & Thakare, V.M. 2011. A systematic study of software quality models. *International Journal of Software Engineering & Applications*, *2*(4), p.61.

Turk, D., Robert, F. & Rumpe, B. 2005. Assumptions underlying agile software-development processes. *Journal of Database Management (JDM)*, *16*(4), pp.62-87.

Turk, D., France, R. & Rumpe, B. 2014. Limitations of agile software processes. *arXiv preprint arXiv:1409.6600*.

Turner, J.H, 1986. The theory of structuration.

Turner, S.P. 2018. *The Social Theory of Practices: Tradition, Tacit Knowledge and Prepositions*. John Wiley & Sons.

Tuteja, M. & Dubey, G. 2012. A research study on importance of testing and quality assurance in software development life cycle (SDLC) models. *International Journal of Soft Computing and Engineering (IJSCE)*, *2*(3), pp.251-257.

Twum-Darko, M. 2014. Factors influencing readiness for transformational e-Government: A perspective of Local Governments in South Africa. *Journal of Public Administration*, *49*(2), pp.512-523.

Twum-Darko, M. 2014. Sustainable local economic development: the role of informatics in determining municipal revenue management.

Twum-Darko, M. & Iyamu, T. 2015. Politicking information technology strategy in organisations: a case study of a selected organisation in South Africa.

Venkatesh, V., Brown, S.A. & Sullivan, Y.W. 2016. Guidelines for conducting mixed-methods research: An extension and illustration. *Journal of the Association for Information Systems*, *17*(7), p.435.

Vosloo, J.J. 2014. *A sport management programme for educator training in accordance with the diverse needs of South African schools* (Doctoral dissertation).

Yao, Y.H. & Lee, H.K. 2004. Applying ISO 9001 and CMMI in Quality-Oriented Knowledge Management for Software Process Improvement. *IJEBM*, *2*(2), pp.140-151.

Wanyama, I. & Zheng, Q. 2010. Organizational culture and information systems implementation: A Structuration Theory perspective. In *Information and Financial Engineering (ICIFE), 2010 2nd IEEE International Conference on* (pp. 507-511). IEEE.

West, D., Grant, T., Gerush, M. & D'silva, D. 2010. Agile development: Mainstream adoption has changed agility. *Forrester Research*, *2*(1), p.41.

Whittington, R. 2010. Giddens, structuration theory and strategy as practice. *Cambridge handbook of strategy as practice*, pp.109-126.

Zainal, Z. 2007. Case study as a research method. *Jurnal Kemanusiaan*, *5*(1).

Zhang, G., Ye, H. & Lin, Y., 2014. Quality attribute modeling and quality aware product configuration in software product lines. *Software Quality Journal*, *22*(3), pp.365-401.

Zimon, D. & Malindžák, D. 2017. Proposal of quality management and technology model supports a subsystem of manufacturing logistics. *LogForum*, *13*(1).

Zope, N.R., Nori, K.V., Kumar, A., Lokku, D.S., Natarajan, S. & Nistala, P.V. 2016. Quality management and software process engineering. In *Software Quality Assurance* (pp. 63-83).

APPENDIX A: CONSENT LETTER

7 Ramir Hof, 13th Avenue
Boston
Bellvile, 7530
021-712-6457
www.seudobit.com

Seudobit

We are one of the world's leading experts in software development and advertising.

23, August, 2017

Permission to conduct survey and collect data at Seudobit

Dear ENCILIAH CHIPUNZA,

Management of Seudobit (Pty) Ltd has granted you permission to conduct your research and survey process with the individuals in the Company. Please remember we have a strict Confidentiality policy (refer to Employment Contract) and therefore ensure that you abide by this agreement.

We wish you all the best with your research and hope that you make every success with your studies.

Kind Regards,
Patience Mpofu
Chief Technology Officer
patience@seudobit.com | 060-863-7459

APPENDIX B: ETHICS CLEARANCE CERTIFICATE

P.O. Box 1906 • Bellville 7535 South Africa •Tel: +27 21 4603534 • Email: majamanin@cput.ac.za
Symphony Road Bellville 7535

Office of the Chairperson **Research Ethics Committee**	**Faculty:** **BUSINESS AND MANAGEMENT SCIENCES**

At a meeting of the Research Ethics Committee on 02 November 2017**, Ethics Approval was granted to** Enciliah Chipunza (216274001)**for research activities Related to the MTech/DTech**: Mtech Business Information System **at the Cape Peninsula University of Technology**

Title of dissertation/thesis/project:	QUALITY MANAGEMENT CHALLENGES IN ITERATIVE SOFTWARE PRODUCT DEVELOPMENT OF A SELECTED SOFTWARE DEVELOPMENT ORGANISATION IN CAPE TOWN, SOUTH AFRICA Lead Researcher/Supervisor: Dr M Twum-Darko

Comments:

Decision: APPROVED

Signed: Chairperson: Research Ethics Committee	**02 November 2017** **Date**

Clearance Certificate No | 2017FBREC479

APPENDIX C: INTERVIEW GUIDE

Introduction

- Welcome, putting individuals at ease.
- Introductions (names) and backgrounds.
- Explain the interview process
    - State the interview's purpose: **to explore your opinion and experience on quality management in iterative software product development process in order to learn more about the current process. The outcome of the research can help identify challenges and improve the process.**
    - Explain that there is no right or wrong answer and I am not here to judge you in any way.
    - State that the interview will take 15-20 minutes of their time.

Interview Questions

Section A: What are the factors influencing quality management process of software product development?

- How would you define quality management?
- What aspects would you say could have an influence on the quality management process of software development process?
- How is quality management embedded in the software development process?
- What are some of the challenges do you face regarding the above?
- What aspects would you identify that would make the process improve?

Section B: What are the socio-technical processes embedded in the quality management process of software product development?

- What technologies do you use (hardware and software systems) in the software development process?
- How is the team structured?
- What sort of sitting arrangement in the office?
- What are your working hours?
- How often do you go for training (internally and externally)?
- What rules and regulations are embedded in the software development process?
- How do you interact with other teams working on the same product?

Section C: What contributes to the ineffectiveness of software product development processes?

- What is the current development process you follow when developing products?
- What are some of the challenges you face?
- To what extent do you think this impacts customer satisfaction?
- Who are the people involved?
- How often do you deploy new functionality?
- How do you interact with the customers?
- How many products do you work on?
- What aspects would you identify that would make the process improve?

Conclusion

You have given us some valuable information. As our time is up, is there anything else you would like to add?

APPENDIX D: DATA TRANSCRIPTS

Date: 5 January 2018
Interviewer: Enciliah Chipunza
Interviewee: AJ: Development Manager

Enciliah: Before we begin the interview itself, I'd like to confirm that you have read and signed the informed consent form, that you understand that your participation in this study is entirely voluntary, that you may refuse to answer any questions, and that you may withdraw from the study at any time.
AJ: Yes, I have read it and understand this.
Enciliah: First question is, what is quality management from your understanding?
AJ: QM is what the tech team builds matches with what's required. Functionality works. Meets expectations. Checklist yes/no quality management / QA is making sure that the right thing is built
Enciliah: Next question is what aspects would you say could influence the QM process?
AJ: What aspects would you say could influence the QA process, badly or well, QA process is (pauses and thinks for a few seconds) is only has good as requirements put on it. So because QA is reactive not necessarily a function which should be part of the design and requirements. QA is just ticking the boxes and if those boxes are accurate then QA is successful. If they are inaccurate even if QA does its job as well as it can its meaningless because the relationship between what the software needs to do and what is **a**rticulated from QA is not what it could be. The success or failure of QA actually the success or failure of a project even if QA is successful if the requirements put on QA or the requirements of the software, the definition of what needs to be done is not well defined or not clear QA can do the best job in the world, but the project will fail.
Enciliah: Next question is what are some of the challenges that you face in the software development process in general and with regards to quality management?
AJ: It's a difficult question. In our world our biggest problem is that [em] there is many many different components- pieces of software that make up our technology stack. It's a legacy thing. So over the time this software stack growing there is many developers doing many different pieces of work largely independently because of the work from home culture that we have and these become products or product systems. So that is a two fold issue because from a development perspective- talk about taking or getting things live. From a development perspective having such a system that is disparate yet so linked to each other means that every developer needs knowledge of systems as much as possible as opposed to being knowledgeable of their own section and ignore everything else. For the business-critical stuff everyone needs to know more of what happening in the whole stack and that's not necessarily

a bad thing, but it does introduce dependencies. If we look at a new product we want to put on our menu, [ah] it's not a simple exercise because of the way the software has evolved. The way the software has evolved there is a lot of dependencies between the product and the systems. If there is a gender type change for example, all product systems would need to be changed. Anyone who wants to support or look after the system needs to know that A, lives in system B, and B lives in system C. All these systems have this information. A developer needs to know how to manage these different systems. That spills over to QA because QA even though their job is to tick the boxes, they need to know ideally, they should be told to look there and there but it becomes part of their thing to know where to look. They get dragged into the interconnectedness of our systems. But we are trying to fix that to maintain a single version of the truth in the different systems so it's a lot easier for people to maintain and make changes and test and QA not to test the entire interconnected systems. Makes things easier.

Enciliah: Next question is, how often do people go for training in this organisation?

AJ: I haven't been to any training since I joined this organisation mmm QA guys I think they go every now and then. I think there have been to one or two. Specifically, in QA context I think there have been some short courses that they have done. But mmm we could do more. So, if we are looking for do we do enough training, no we don't.

Enciliah: The next question is, are there any quality management rules and regulations that needs to follow in the software development process?

AJ: Clarify please

Enciliah: For example, code reviews

AJ: There are no code reviews as far as I know. The process we have is we get requirements that get scheduled into a sprint, gets developed and QA tests that through their checklist of what is the successful implementation or requirement. They do this in Dev environment and once deployed they test in live. There is no regulation it's just a process.

Enciliah: Are there any processes in place to ensure that the customer is satisfied?

AJ: Developers do not interact with the customer. They shouldn't. The product person is the one who interacts with the customer. The product person originates the requirements to make the customer's life easier. He/she represents the customer and at best that's the one the developers will interact with.

Enciliah: Last one, what do you think will help improve the management of quality in the software product development process?

AJ: [ah] I think we are improving but more attention to detail, more efficiencies in the process, more planning, collaboration. Just understanding what needs to be done. Questioning that needs to be done. Not questioning as such but just understanding and getting people involved to talk to each other i.e I know you want this to be done but what about if we do it this way or do you know we do it this would impact this and this and will take 5 times longer than if we do it this way. So that kind of upfront collaboration to ensure that when we get to whatever the

ground everyone is happy with what's done. That's what we try and do in our Monday sprint meeting, but we just need to do a little bit more.

Enciliah: Do you think one meeting is enough?

AJ: We have more than one meeting. There used to be only one but now we have another Monday meeting where developers come in and sit with the product team to discuss what I mentioned earlier. This way is better. For example, they will discuss, this is how it should work, this is how long it will take. We also have our planning meeting as well as the Monday every 2 weeks prioritization meeting. Understanding what needs to be done, that collaboration around priority is happening on Monday.

Enciliah: Are there any retrospectives after the 2 weeks sprint?

AJ: Ah- no nothing formal. We do agile not follow a specific methodology such as scrum or Kanban. Agile is breaking tasks into smaller realistic measurable tasks. Try get the best impact in the smallest amount of time by modulising what you need to do.

Enciliah: Is there anything else that you would like to add?

Participant: Nope, I think that about covers it.

Enciliah: Well, thanks for taking the time to talk with me today. I really appreciate it.

Date: 9 January 2018
Interviewer: Enciliah Chipunza
Interviewee: MJ: UX Designer/ Product Manager

Enciliah: Before we begin the interview itself, I'd like to confirm that you have read and signed the informed consent form, that you understand that your participation in this study is entirely voluntary, that you may refuse to answer any questions, and that you may withdraw from the study at any time.
MJ: Yes
Enciliah: First question is, what is quality management process from your understanding in this organisation?
MJ: When I started we didn't have any process in place until when one QA member joined. Since he joined the infrastructure we are putting in place around now and the stuff they are doing is good and I think we are in the right track to a good quality check process. If you asked me the same question last year I would have said something different.
Enciliah: How often are bugs found in production?
MJ: Umm, it doesn't happen as much as it used to be, customers used to find bugs in production and call the call centre a lot. Not ideal.
Enciliah: Are you involved in testing yourself?
MJ: If it's I own for example the FS app, I make sure that I test thoroughly as well as QA yeah. I get QA to test it and then I do that on my side. Yeah
Enciliah: What level of testing is done for example manual or automated testing?
MJ: I don't know, it's all maul testing and the testing done umm every set of functionalities you can observe. Making all the functionality works like they are supposed to be. Also, do accepted behavior testing. Which is you input symbols and special characters in username field as an example where as it should be only letter type of thing and that type of testing in place and how that type of functionality reacts to that type of thing. Testing is always what you see in functionality and expected behavior and then the unexpected behavior.

Enciliah: So how long have you been working here?
MJ: 3 years
Enciliah: Next question since you joined the organisation have you gone for training inhouse or outside?
MJ: Umm, yes there has been for QA people. Study type of QA but would want to be given the opportunity be able to do courses that would benefit the organisation. What training does is that it gives you a little bit of certainty that what you are doing is right. It deferres your decision to someone else that this is the right thing you are doing as opposed to lambering around and figuring out things from mistakes, which isn't a bad thing because it gives us homework to work

with I think.

Enciliah: Any suggestions for improvement for the software delivery process?

MJ: Ummm, I think you can't take shortcuts. What happens is that if QA comes with a list of bugs then developers will go in and fix them and create a new version. The problem is they will just test those 10 bugs to make sure that its fine. But, when the fixes are done then something else could have broken so its repetitive but needs to be thorough and that's what we need to do. As soon as a new version is released by the developers.

Interviewer: Do you have anything else you would like to add?

MJ: No, cool

Enciliah: Thank you for your time.

Date: 10 January 2018
Interviewer: Enciliah Chipunza
Interviewee: BG: Senior Developer

Enciliah: Before we begin the interview itself, I'd like to confirm that you have read and signed the informed consent form, that you understand that your participation in this study is entirely voluntary, that you may refuse to answer any questions, and that you may withdraw from the study at anytime.
BG: Yes
Enciliah: How would you describe quality management in the software development process?
BG: Actually, it's a two fold process, what happens is when developers finish their work they give it to QA. QA just does quality assessment. The actual overall usability of the code. The actual working of the code. The product people have to take ownership because if they don't some of the functionality may be missed by QA people. So, QA qualifies to a certain extent and then product has to check to see what has been developed in line with what is expected in line with their processes at the beginning.
Enciliah: What are some of the challenges currently being faced in the process?
BG: mmm, I don't think necessarily we have any challenges but with all processes there is room for improvement. So, you will see since we are agile, you find in each and every organisation they want to tailor made it so that it actually suits their environments. So testing can be sort of improved or enhanced. If ah QA decides to incorporate use of tools that makes their jobs easier. That's the only thing that lacks at SasTech. But I'm saying it all depends on the company if they want to use tools or not. Some says tools may slow down their processes and others say it may speed up the process. So, it is just a matter of opinion most of the time.
Enciliah: What are some of the tools you are using- hardware/software?
BG: We use almost everything in terms of software we primarily use .Net because most of our portals are .Net and we have some backends using Java and C. But would also prefer some of the new technologies that other companies are now using.

Enciliah: How is your team structured?
BG: We have one manager at the top- team leader and sub team leaders and we have other devs like Ken with devs under him. Then QA and has no team lead the report to the manager at the top.
Enciliah: Next question is how often do people go for training in this organisation?
BG: Ah, I haven't seen any for devs. But I think there are plans. Its needed, it would be nice to learn new technologies. As a developer you need to constantly improve your technical skills all the time. Yes, search, try things so that you are familiar with what people are using out there. Like for example, if you tell people we are using Java. They will say, what? You're

ancient. Because most people now they use languages like GO and Scala because those are modern language so yeah as developer if you don't, you are left behind. Besides stuff happening at work you must go out and look for new tech on your own.

Enciliah: What are some of the things that can improve with regards software product development process here?

BG: [ah] like any software organisation out there, there is always room for improvement. What happens with these things here, the moment a new person walks in they discover issues that lack and they try to suggest, guys lets use this language it becomes better or lets use this tool so at times it the leaders at the top that don't want change because the issue is they are old school type of guys so if you change to a language like Go they don't even know it and the learning curve for them is very tough. Exactly that they don't want. But in the long run these types of changes help. Because tools and software help keep up with tech.

Enciliah: Are there any rules, regulations, standard practice?

BG: Like any organisations you cannot put stuff on the server without testing it. We use a tool so that every developer uses a common language and standards when you put in the repository it should work for everyone

Enciliah: That's it

Enciliah: Is there anything else that you would like to add?

BG: Nope

Enciliah: Well, thanks for taking the time to talk with me today. I really appreciate it.

Date: 11 January 2018
Interviewer: Enciliah Chipunza
Interviewee: NK: Developer

Enciliah: Before we begin the interview itself, I'd like to confirm that you have read and signed the informed consent form, that you understand that your participation in this study is entirely voluntary, that you may refuse to answer any questions, and that you may withdraw from the study at any time.
NK: Yes, I had read it and understand this.
Enciliah: How would you describe software quality management at your company?
NK: Good enough
Enciliah: What rules and regulations are embedded in the software development process at your company?
NK: No tests, no deployment.
Enciliah: What are some of the quality management challenges in the software development process?
NK: Writing tests can be time consuming
Enciliah: How can the process be improved?
NK: Use of automated testing
Enciliah: Next question is, are you involved in testing and what level of testing is done?
NK: Brief testing after development work is done
Enciliah: What processes are in place to ensure that customer satisfaction is met with each iteration.
NK: Product managers visit client sites or to the call centre when deployments are performed so they have first-hand experience of challenges faced
Enciliah: Is there anything else that you would like to add?
NK: No, not really
Enciliah: Well, thanks for taking the time to talk with me today. I really appreciate it.
Date: 12 January 2018
Interviewer: Enciliah Chipunza
Interviewee: RG: Developer

Enciliah: Before we begin the interview itself, I'd like to confirm that you have read and signed the informed consent form, that you understand that your participation in this study is entirely voluntary, that you may refuse to answer any questions, and that you may withdraw from the study at any time.
RG: Yes, I have.
Enciliah: First question is, what is quality management from your understanding?

RG: Defining standards by which quality needs to be measured

Enciliah: Next question is what aspects would you say could influence the QA process?

RG: Management buy in, developers chase new things, and don't care, when other priorities trump up. skills and experience of developers will ensure there are less rules

Enciliah: How is quality management embedded in the software development process?

RG: There are no formal rules, depends on team and individuals, better design well collaborated between business and development in designing a solution is key. Design, Planning, Making time and Code reviews.

Enciliah: What technologies are you currently using (hardware and
software systems?

RG: Swift iOs mobile platform, Rehive backend packaging

Enciliah: Next question is how often do people go for training in this organisation?

RG: There is a lot of inter-learning

Enciliah: What are some of the challenges and things that can improve?

RG: Lack of team coherences, Lack of design, design workflow, lots of testing

Enciliah: Anything else you would like to add?

RG: Nothing

Enciliah: Okay, thank you for taking the time to talk with me today. I really appreciate it.

Date: 16 January 2018

Interviewer: Enciliah Chipunza

Interviewee: JT: Developer

Enciliah: Before we begin the interview itself, I'd like to confirm that you have read
and signed the informed consent form, that you understand that your participation in this study is entirely voluntary, that you may refuse to answer any questions, and that you may withdraw from the study at any time.

JT: Yes

Enciliah: First question is, what is quality management from your understanding?

JT: Same as QA, is to ensure that everything works as intended and when changes are made it still works

Enciliah: Next question is what aspects would you say could influence the QA process?

JT: Deadline can affect the time it takes for quality to be assessed, experience of the team or the team lead, depth of testing

Enciliah: What aspects would you identify that would make the process improve?

JT: Creating a process in the first place, making sure there is depth testing, fully planned testing, on how to test, and defining intended outcomes, consistent testing, assessment of what hasn't been working on- depth testing, implementing automated testing.

Enciliah: Next question is, what technologies are you currently using (hardware and
software systems?

JT: Linux, sp.net, html, JavaScript,

Enciliah: Next question is how often do people go for training in this organisation?

JT: On the job training lots of it

Enciliah: What rules and regulations are embedded in the software development process?

JT: confident on changes, test properly your code, foresight not directly changing, think properly, write unit tests, automated testing,

Enciliah: What processes are in place to ensure that customer satisfaction is met with each iteration?

JT: product owner does the testing

Enciliah: What aspects would you identify that would make the process improve?

JT: fully specced features

Enciliah: How often are they picked up and time frame before they get noticed?

JT: Most of the time

Enciliah: Are they are any code reviews or pair programming?

JT: None involved in

Enciliah: Anything else you would like to add?

JT: No

Enciliah: Thank you for taking the time to talk with me today. I really appreciate it.

Date: 17 January 2018
Interviewer: Enciliah Chipunza
Interviewee: TK: Senior Developer

Enciliah: Before we begin the interview itself, I'd like to confirm that you have read and signed the informed consent form, that you understand that your participation in this study is entirely voluntary, that you may refuse to answer any questions, and that you may withdraw from the study at anytime.
TK: Yes
Enciliah: How would you describe quality management at this organisation?
TK: In my experience at least the development I have worked on its been present because everything that I have worked ummh let me just say in my experience it's been therein as far as my own technical testing and once I am done with that I let the product manager know and they pass it to QA and then QA do what they do and we have release form saying these are the test cases against the product release. QA does pass/fail to all the test cases. And then we go back and forth with whoever we are integrating with to iron out anything that has failed in the whatever I am supposed to do. Then re QA it to check if everything passes that needs to pass before certified so to speak. So that's for everything I have which went through QA process however, I do have some experience where something went through QA and we were now taking it live and when we put on pre-production we noticed a big issue which hadn't been picked up by QA. QA had believed everything is fine. To put it in perspective the error was with regards to a hard-coded amount which could have resulted in the company losing a lot of money and wasn't caught by QA.
Enciliah: Next question is how often are bugs found in production?
TK: In production ummmh
Enciliah: Less time or more times
TK: My apps not as yet, the first big project is only going live now. But I know there are issues that have been escalated to other developers found in production. They have been asked to look into something that then it's hard to say it's a bug because in the sphere of development where a developer is the contact person rarely acknowledges it's a bug. I noticed a bug and the developer said its information being sent from the other end so it's always like that. Because of the grey area where nobody sees what's really going on. I have also seen issues come up because the partner changed something on their end.
Enciliah: How is your team structured?
TK: Immediate team is small it's me and my manager, so I guess as whole overview we have another manager at the top
Enciliah: How often does your team meet?
TK: Sprint meeting every two weeks with all tech stakeholders link up on what everyone is

doing.

Enciliah: Before and after deployment do people, meet to discuss what's coming in the release and after the release?

TK: After release, umm once again I haven't had stuff go live. I guess so but may not be official. With my team release go with my team the stuff goes into a pre-server and is open to a few users and there some monitoring that happens. People will watch and communicate and then it goes live and once live there just normal monitoring that support does and reports/escalate if there are issues.

Enciliah: So how are requirements fed to the team?

TK: Requirements come from the product manager and my area manager are typically in the meeting and support also is in the meeting with prospective clients. Once the client has been engaged then Tech is invited to join in some of the meetings. I get specifications from my manager. When he was not around then I went into those meetings which he normally goes to.

Enciliah: Do people for training in/out?

TK: I have never come across it and there are cases when I have thought training would be nice when you find yourself working on new stuff. It was because I spent hours figuring out things.

External, I don't know who has taken it but is on offer and is communicated. Like if you want to do anything that enhances you, or your job functions the company will pay.

Enciliah: What do you think will help improve the management of quality in the software product development process?

TK: Mmm I think one area, I would say is from the dev perspective is to be in those meetings when product decides what needs to a sprint or meetings with business partners ift would useful for the person developing to be in there. Otherwise you might find yourself building something you have no background info about. When these meetings happen, there are a group of people and someone is going to ask a technical question but if tech is present then you are going to say but our system works like this other than them using a spec. Chances are that you would miss some of the issues discussed.

From the basis of QA, I would say, we could have more people testing QA tester A and QA tester both going through the same tests.

Enciliah: Are there any code /peer reviews?

TK: No, official our dev here is a one man show. If you are a dev in a particular product then its all you so there is not a lot of collaboration. I don't know if it's probably because of not having a product that's huge where multiple people needed and multiple moving parts but yeah for typically one person speaks the cards for that. Even if in our case it could be manager being called out for those cards but when I am the one working on some of those cards.

Enciliah: That's it, is there anything else that you would like to add?

TK: Nope

Interviewer: Thanks for taking the time to talk with me today. I really appreciate it.

Date: 17 January 2018
Interviewer: Enciliah Chipunza
Interviewee: TC: Developer

Enciliah: Before we begin the interview itself, I'd like to confirm that you have read and signed the informed consent form, that you understand that your participation in this study is entirely voluntary, that you may refuse to answer any questions, and that you may withdraw from the study at anytime.
TC: Yes
Enciliah: How would you describe software quality management at your company?
TC: Quality is managed, umm I think it's not managed well here especially testing. There is no standard way of coding. There is need to communicate the standards which must be followed.
Enciliah: What rules and regulations are embedded in the software development process at your company?
TC: It's just that when you are a developer you must know to test your code against requirements before passing to QA. Ask questions to understand requirements. You do test and then do until it works.
Enciliah: How can the process be improved?
TC: Regular meeting to get updates on what everyone is doing. Number 2 put standards in place because these standards bring quality to the system and better engagement with QA. There is no automated testing. Automated testing for QA, they will be able to run some code. It nice because it catches some bugs that might be missed by manual testing.
Enciliah: How often does your team meet?
TC: It's not agile perse but we try to meet daily with product manager and manager and product managers are always there when I have questions.
Enciliah: How is your team structured?
TC: There is no clear structure for me. I don't even know what some of the people I work with do.
Enciliah: That's it
Enciliah: Is there anything else that you would like to add?
TC: Nope, I think that about covers it.
Enciliah: Thanks for taking the time to talk with me today. I really appreciate it

Date: 18 January 2018
Interviewer: Enciliah Chipunza
Interviewee: BC: Tester

Enciliah: Before we begin the interview itself, I'd like to confirm that you have read and signed the informed consent form, that you understand that your participation in this study is entirely voluntary, that you may refuse to answer any questions, and that you may withdraw from the study at anytime.
BC: Yes
Enciliah: How would you describe software quality management at your company?
BC: QA starts from the get-go. When product guys have new requirements and have been developed and we as QA are available to test. But it's difficult to have a system with 100% bug free
Enciliah: How are you involved in testing and what level of testing is done?
BC: We receive a spec from the product managers, we build test cases, we test and get outcomes of the tests. If pass we send to product managers for sign off if there are bugs we send back to developers.
Enciliah: How can the process be improved?
BC: Proper collaboration- Product managers, QA and Devs all in one room. Get everyone onboard so QA is involved from the get-go so we are aware of what's coming.
Enciliah: How often does your team meet?
BC: Every day for 10-15 mins. We discuss what everyone is working on, challenges and advised who we need to speak to
Enciliah: How is your team structured?
BC: The IT manager and 2 senior testers and one junior tester
Enciliah: How often are bugs found in productions?
BC: A lot and most of them are beyond QA because when we test the things will be working but some changes are done that we are not aware before release and you will later hear someone say it was me a week later and because of developers working from home it's difficult to collaborate.
Enciliah: Automated testing?
BC: No automated testing because the systems are too old for automation. We do manual testing and use software like Postman
Enciliah: Is there anything else that you would like to add?
BC: Nope, I think that about covers it.
Enciliah: Well, thanks for taking the time to talk with me today. I really appreciate it

Date: 19 January 2018
Interviewer: Enciliah Chipunza
Interviewee: TA: Product Manager

Enciliah: Before we begin the interview itself, I'd like to confirm that you have read and signed the informed consent form, that you understand that your participation in this study is entirely voluntary, that you may refuse to answer any questions, and that you may withdraw from the study at anytime.

TA: Yes

Enciliah: First question, how would you describe software quality management at your company?

TA: So, it is difficult because there many moving parts so it's difficult to have proper end quality inspection. Also, here there is no proper environment for QA if there was I would say we would have quality products. Also, the nature of the business. It takes months to get approval from regulators and with third parties so when it comes back it's like we have to do it now, so testing is done, and the quality is hard to maintain. It's always a trade-off whether you want to release something 100% done or 80% done. It's always evaluated here in terms of revenue.

Enciliah: What are some of the QM challenges?

TA: The nature of the business because that also determines speed, deadlines also because of 3rd parties involved might threaten the quality because you can't always stick to deadlines when working with other parties. The other challenge is resource constraints, we don't have the luxury of a huge dev team.

Enciliah: Any rules and regulations?

TA: In terms of

Enciliah: For quality management in the development process?

TA: Yes, we do change control before taking something live, that change control is signed off by finance, risk, executives. People who have different insights of the business sign it off. Even before then you will call in a finance person and devs. Here is what I see, where are the big problems. We do have 2 controls- change control and something called loop.

Enciliah: Are you involved in testing?

TA: All product managers are involved in testing their products. You will test with QA. QA is only 2 people. What product managers will do, will give QA my own test pack with what needs to be tested. So we do it in 2 ways- usability and functionality

Enciliah: Any training?

TA: The company encourages as to do that, but it is on the product managers to go and say this is where I want to improve. There are no strict rules that the product manager must go for training. You also have to learn from your team.

Enciliah: What can be done to improve the process?

TA: To have a proper QA environment. There is challenge you can test something in QA but you put on a device and it doesn't work. Change roll out plans like versions having continuous integration and version control system, so you can see what changed in that last version. Revenue vs new stuff(architecture) there is always that trade off.

Enciliah: Is there anything else that you would like to add?

Participant: No.

Enciliah: Well, thanks for taking the time to talk with me today. I really appreciate it

Date: 23 January 2018
Interviewer: Enciliah Chipunza
Interviewee: LT: Developer

Enciliah: Before we begin the interview itself, I'd like to confirm that you have read and signed the informed consent form, that you understand that your participation in this study is entirely voluntary, that you may refuse to answer any questions, and that you may withdraw from the study at anytime.
LT: Yes
Enciliah: First question, how would you describe software quality management at your company?
LT: Talking about the quality umm we think about how the app is structured because based on the structure of the app you can think about how the process should be and how the process should be and what quality aspects you need to measure. It's like you need to have a, to set up a rule or formula to feed the app. And then you know what you need to measure, set up KPIs for yourself and you need to first give it a target and then how to measure it and then give it the expected results and after a while you know what your point is. To see the quality, it seems you almost have to know what you want, whatever target and how to measure i.e manager says they want app to run faster and you need to specify how fast you want. What experience you want to give the users something you need to think about. The apps shouldn't crush so you must use stabilisers.
Devs- product managers and design need to be involved. At times bugs are introduced because of bad design and sometimes a bug happens because the business give has some issue or like sometimes a bug happens because the business wants too much functionality and the dev puts everything in and the app is slow for the user. That's a common case.
Enciliah: Are you involved in testing?
LT: I like to, before I submit my code change or mark my ticket to be ready for testing. I test it and think of edge cases- some workflows the tester night not be aware of so, I write down the test cases for them.

Enciliah: Any training?
LT: Not since I joined
Enciliah: What can be done to improve the process?
LT: You can always improve the process, but I don't think you can solve that. For our company and every company has a legacy issue. Bugs are always there and people knowing they are bugs but ignore them so if you want to improve start from scratch and then you have an opportunity to think in white paper to hold insights and there are no legacy issues and you have option to use new technologies. Some languages don't allow you to use. And to always

check for null pointers.

Enciliah: That’s it

Enciliah: Is there anything else that you would like to add?

LT: No

Enciliah: Well, thanks for taking the time to talk with me today. I really appreciate it

Date: 24 January 2018
Interviewer: Enciliah Chipunza
Interviewee: LI: Product Manager

Enciliah: Before we begin the interview itself, I'd like to confirm that you have read and signed the informed consent form, that you understand that your participation in this study is entirely voluntary, that you may refuse to answer any questions, and that you may withdraw from the study at anytime.
LI: Yes
Enciliah: First question, how would you describe software quality management at your company?
LI: The most important thing in this process is communication. Precisely deciding what is standard practice.
Enciliah: Are there any rules and regulations set in the process?
LI: The rules I would say is deciding who the people are that needs to be involved and which part of the product. So, when something is changed in the product or section of the product at least you know who are involved and responsible for that change.
Enciliah: Any training?
LI: Not enough
Enciliah: How often are bugs found in the software product?
LI: At least once a week
Enciliah: Are you involved in testing?
LI: Sometimes
Enciliah: How often does your team meet?
LI: Once every two weeks formally

Enciliah: What are some of the challenges that are faced??
LI: Time always is an issue and it forces to prioritise, and this is not done properly
Enciliah: That's it
Interviewer: Is there anything else that you would like to add?
Participant: No
Interviewer: Thanks for taking the time to talk with me today. I really appreciate it.

Date: 24 January 2018
Interviewer: Enciliah Chipunza
Interviewee: BS: Developer

Enciliah: Before we begin the interview itself, I'd like to confirm that you have read and signed the informed consent form, that you understand that your participation in this study is entirely voluntary, that you may refuse to answer any questions, and that you may withdraw from the study at anytime.
BS: Yes
Enciliah: How would you describe software quality management at your company?
BS: Not very thorough
Enciliah: Are there an rules and regulations set in the process?
BS: All bug fixes and features get tested by QA and customer sort before release.
Enciliah: Any training?
BS: Not enough
Enciliah: What processes are in place to ensure customer satisfaction?
BS: Surveys and customer support staff
Enciliah: What are some of the challenges?
BS: Testers don't fully understand the feature being developed. Developers not comprehending all the side effects that a new feature or fix brings. Product team not giving developers specifications that are considerate to other parts of the system.
Enciliah: Are you involved in testing?
BS: Yes, unit testing and testing as I develop or fix something
Enciliah: What can be done to improve the process?
BS: I think if the product team offers speculations that are well specified with use cases for QA to test there will be less reliance on the developer and QA will have less pressure to know the business logic if product gives it

Enciliah: That's it
Enciliah: Is there anything else that you would like to add?
Participant: No, not really
Enciliah: Thanks for taking the time to talk with me today. I really appreciate it

Date: 25 January 2018
Interviewer: Enciliah Chipunza
Interviewee: TJ: Developer

Enciliah: Before we begin the interview itself, I'd like to confirm that you have read and signed the informed consent form, that you understand that your participation in this study is entirely voluntary, that you may refuse to answer any questions, and that you may withdraw from the study at anytime.
TJ: Yes
Enciliah: How would you describe software quality management at your company?
TJ: We get given specs and then we make sure that, mmm code and then make sure its quality. Whenever, you have a problem you can communicate with product and other developers so that you are not left behind. But problems do occur.
Enciliah: How often are bugs found in production?
TJ: A few
Enciliah: What challenges?
TJ: On handing over of work. And then all of a sudden and you have started working requirements change it disturbs the whole process
Enciliah: Are you involved in testing?
TJ: What I do if I find an issue, I debug and fix and give it to QA to test.
Enciliah: What can be done to improve the process?
TJ: Actually, I think ah, I can just say if you want to make sure that requirements are understood before work is done. This will avoid issues along the way.
Enciliah: How is your team structured?
TJ: It is changing. It depends on the product you are working on. The other team is working on this product and others on another that's how is structured.

Enciliah: Is there anything else that you would like to add?
TJ: No
Enciliah: Thanks for taking the time to talk with me today. I really appreciate it

Date: 25 January 2018
Interviewer: Enciliah Chipunza
Interviewee: FD: Developer

Enciliah: Before we begin the interview itself, I'd like to confirm that you have read and signed the informed consent form, that you understand that your participation in this study is entirely voluntary, that you may refuse to answer any questions, and that you may withdraw from the study at anytime.
FD: Yes
Enciliah: How would you describe software quality management at your company?
FD: You see there is management of quality because we always start a project from something like requirements and we build from that and make sure we have quality throughout
Enciliah: How often are bugs found in production?
FD: A few
Enciliah: What challenges?
FD: Information, lack of information and also sometimes something is signed off but some sure may arise from things that were previously signed off
Enciliah: Are you involved in testing?
FD: We are not involved in testing. We do our own tests like unit tests but at the moment we are not doing that. The only testing that happens here is manual by QA. So its just testing for bugs stuff like that. As developers we are supposed to do unit tests.
Enciliah: How often are bugs found in production?
FD: There are always bugs. I would say from my experience 10%-20% but maybe mostly UI related or incorrect responses.
Enciliah: What can be done to improve the process?
FD: First off getting an understanding of what we are going to build and getting relevant information in order to start planning how we are going to deliver the project. So a backlog with user stories that needs to be created. I think we need a proper backlog, do research around it and additions to the original spec we add to the backlog finish off next phase we include it. That's being agile
Enciliah: Are there any code reviews?
FD: No code reviews. There are supposed to be code reviews, but we don't do that.
Enciliah: Is there anything else that you would like to add?
FD: No
Enciliah: Thanks for taking the time to talk with me today. I really appreciate it

Date: 30 January 2018
Interviewer: Enciliah Chipunza
Interviewee: JK: Tester

Enciliah: Before we begin the interview itself, I'd like to confirm that you have read and signed the informed consent form, that you understand that your participation in this study is entirely voluntary, that you may refuse to answer any questions, and that you may withdraw from the study at anytime.
JK: Yes
Enciliah: How would you describe software quality management at your company?
JK: Right now, I feel there is no proper process in place. It's always adhoc stuff. At any given time, it's all about can you do this, can you do that. No quality management of the software. I think it could be improved. There is too much pressure from all sides to get things out that affects the quality of the product. It's always tight timelines. There is something that more urgent than the other thing. So, we need to streamline that. I can't really say there is a good quality process in place.
Enciliah: How often are bugs found in production?
JK: A few
Enciliah: What challenges?
JK: Information, lack of information and also sometimes something is signed off but some sure may arise from things that were previously signed off
Enciliah: How are you involved in testing and what level of testing?
JK: Test on devices and am very strict. I test everything sometimes we have to roll out something urgently. You can't really catch all the bugs, but you test to the best of your ability.
Enciliah: Rules and regulations?
JK: Its not from us really or for us it's because of the urgency from outside to get things out. But our rules give us a clear guideline of what needs to be tested in the beginning. We need to be involved throughout the lifecycle of development. From the planning stages that could streamline everything. Very few people involve us that would be efficient if everyone did because we can test all the stages of the lifecycle of the project/product.
Enciliah: What can be done to improve the process?
JK: Umm, better test procedures, documentation it would make a big difference and could affect the quality of the process and clear guidelines. As QA we have a process in place we follow of our own that works pretty well. We don't always get documentation. There are some people who do it, but others don't. It's very efficient to have documentation because now you have something to work with, so I think that is very very important.
Enciliah: Are there any code reviews?
JK: No code reviews. There are supposed to be code reviews, but we don't do that.

Enciliah: That’s it

Enciliah: Is there anything else that you would like to add?

JK: No

Enciliah: Thanks for taking the time to talk with me today. I really appreciate it

Date: 30 January 2018
Interviewer: Enciliah Chipunza
Interviewee: AC: Developer

Enciliah: Before we begin the interview itself, I'd like to confirm that you have read and signed the informed consent form, that you understand that your participation in this study is entirely voluntary, that you may refuse to answer any questions, and that you may withdraw from the study at anytime.
AC: Yes
Enciliah: How would you describe software quality management at your company?
AC: Business comes up with requirements. They might tell you that this was developed 3 years ago, and it has 5 steps and we want to reduce it to 3. As a developer then you would change something for instance I am currently working on a job that was written on the database but now writing It in code so that's an example of a request
Enciliah: What challenges?
AC: For me challenges sometimes depends- sometimes tight deadlines hinder quality. You might end up cutting corners. But it's really something that you cannot control because on who /where the requirements are from. You can always speak to them and say if you want this done, this might get affected we might end up cutting corners there are you sure you want to do this. Let's break it up in phases so let's do phase 1 now and phase 2 if you know what I mean so that you don't come back and change something.
Enciliah: How are you involved in testing and what level of testing?
AC: Basically, you get sprint work and afterwards you do your own testing- unit tests but here yet. Also, we could introduce a test document which developers need to write results of their own testing. This is to compare with production. In cases where the tester might say the change wasn't there, this document will be evidence and contains screenshots of your tests of what the system you are working on. So that when the tester come back you can show your tests. It also helps because they always restore databases so it's kinda helps with argument as well. With testers involvement is to catch issues or bugs and report them with different scenarios they tested so it can be reproduced, debugged and fixed.
Enciliah: Rules and regulations?
AC: It's not from us really or for us it's because of the urgency from outside to get things out. But our rules give us a clear guideline of what needs to be tested in the beginning. We need to be involved throughout the lifecycle of development. From the planning stages that could streamline everything. Very few people involve us that would be efficient if everyone did because we can test all the stages of the lifecycle of the project/product.
Enciliah: That's it
AC: Is there anything else that you would like to add?

Enciliah: No

Enciliah: Thanks for taking the time to talk with me today. I really appreciate it

Date: 31 January 2018
Interviewer: Enciliah Chipunza
Interviewee: TM: Developer

Enciliah: Okay, before we begin the interview itself, I'd like to confirm that you have read and signed the informed consent form, that you understand that your participation in this study is entirely voluntary, that you may refuse to answer any questions, and that you may withdraw from the study at anytime.
TM: Yes
Enciliah: How would you describe software quality management at your company?
TM: Quality management is handed by a QA team during a period known as 'code freeze' every sprint before release.
Enciliah: What are the challenges?
TM: QA often does not have enough resources and time to effectively cover every change to be released. This often leads to delayed releases and rushed testing.
Enciliah: How are you involved in testing and what level of testing?
TM: Yes. Besides the usual unit and integration tests that go with development, we often have to test our features and bug fixes first before it reaches QA.
Enciliah: Rules and regulations?
TM: Software development follows an agile approach, with 2 sprints
Enciliah: How can the process be improved?
TM: This process can be approved by hiring more QA resources and automating more tests.
Enciliah: How do you ensure customer satisfaction?
TM: Critical bug fixes and features are planned for each iteration, to ensure that they are released as quickly as possible.

Enciliah: Is there anything else that you would like to add?
TM: No
Enciliah: Thanks for taking the time to talk with me today. I really appreciate it

Date: 1 February 2018
Interviewer: Enciliah Chipunza
Interviewee: LM: Developer

Enciliah: Before we begin the interview itself, I'd like to confirm that you have read and signed the informed consent form, that you understand that your participation in this study is entirely voluntary, that you may refuse to answer any questions, and that you may withdraw from the study at anytime.
LM: Yes
Enciliah: How would you describe software quality management at your company?
LM: Its make sure that of good user experience. As a developer your job is to provide a good user experience and its obviously difficult
Enciliah: What are the challenges?
LM: Due to deadlines and timelines you can make mistakes
Enciliah: How are you involved in testing and what level of testing?
LM: I do my own testing all the features I have built and people in QA play a big role and its difficult if they don't what to ensure good user experience
Enciliah: How can the process be improved?
LM: In planning you can project what are some of the risks are they. When you get into the process you can identify risks and its effect will be.
Some kind of management system for QM.
Enciliah: Is there anything else that you would like to add?
LM: No
Enciliah: Thanks for taking the time to talk with me today. I really appreciate it

Date: 2 February 2018
Interviewer: Enciliah Chipunza
Interviewee: TP: Product Manager

Enciliah: Before we begin the interview itself, I'd like to confirm that you have read and signed the informed consent form, that you understand that your participation in this study is entirely voluntary, that you may refuse to answer any questions, and that you may withdraw from the study at anytime.
TP: Yes
Enciliah: How would you describe software quality management at your company?
TP: Its a thorough process. There is code freeze week that allows developers to
Enciliah: What rules and regulations are in the quality management process?
TP: Spec, Devs develop, QA tests., release is done
Enciliah: What are some of the challenges?
TP: There is resource constraints for the QA department
Enciliah: How can the process be improved?
TP: Planning well QA informed of work load All code that gets released must be working well
Enciliah: Are you involved in testing and what level of testing is done?
TP: No, not at the moment
Enciliah: What processes are in place to ensure customer satisfaction?
TP: Customer issues are logged with expected outcomes. Devs work on issues with biggest priorities, and those are done first. Not all can be attended to, but the co tried to ensure (especially for high profile companies) that it is done.
Interviewer: that's it, thanks for taking the time to talk with me today. I really appreciate it

Date: 2 February 2018
Interviewer: Enciliah Chipunza
Interviewee: KG: Developer

Enciliah: Okay, before we begin the interview itself, I'd like to confirm that you have read and signed the informed consent form, that you understand that your participation in this study is entirely voluntary, that you may refuse to answer any questions, and that you may withdraw from the study at anytime.
KG: Yes
Enciliah: How would you describe software quality management at your company?
KG: Very good and effective
Enciliah: What rules and regulations are in the quality management process?
KG: 1. Peer reviewing of code for efficiency and best practices will be required for all software 2. All software units are to be tested for golden path(expected) behaviour and edge case testing for reliability 3. All code changes shall require regression testing to be conducted before promoting code to production 4. Load/stress testing to be carried out on production like environments to guide sizing and capacity requirements 5. Extensive UAT testing to be done before sign-off
Enciliah: What are some of the challenges?
KG: Project time constraints often prevent adherence to the quality management processes in full. Another challenge is the lack of big testing teams at client sites which may prevent some issues from being discovered.
Enciliah: How can the process be improved?
KG: Increase testing code coverage and incorporating more testing time in project life cycles. Full time testers also need to be increased
Enciliah: Are you involved in testing and what level of testing is done?
KG: No

Enciliah: What processes are in place to ensure customer satisfaction?
KG: UAT and regression testing is in place to provide customer satisfaction
Enciliah: That's it
KG: Is there anything else that you would like to add?
KG: No
Enciliah: Thanks for taking the time to talk with me today. I really appreciate it

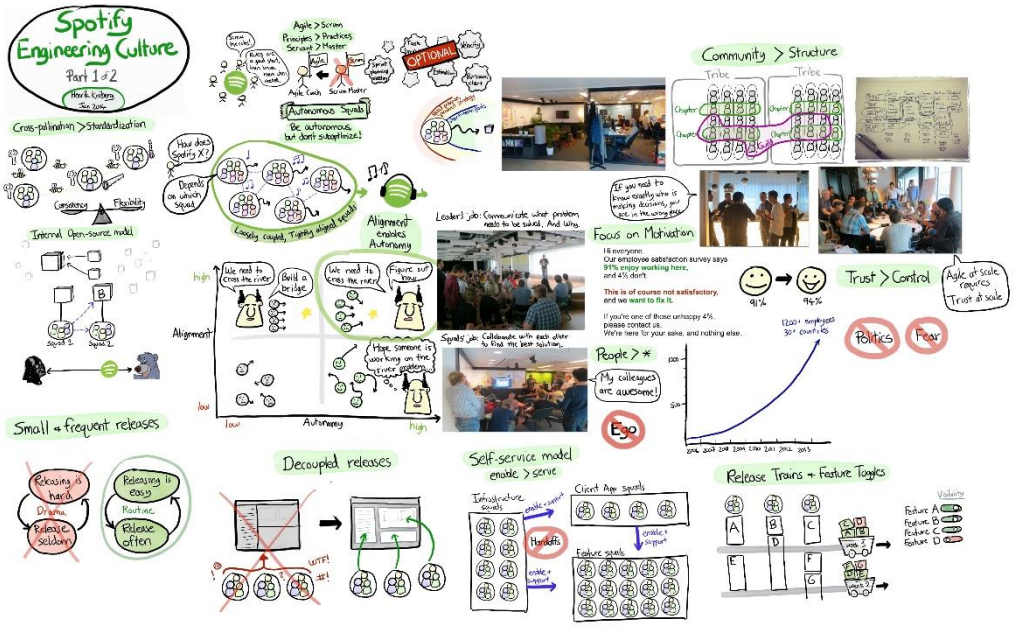
Date: 3 February 2018
Interviewer: Enciliah Chipunza
Interviewee: TO: Product Manager

Enciliah: Okay, before we begin the interview itself, I'd like to confirm that you have read and signed the informed consent form, that you understand that your participation in this study is entirely voluntary, that you may refuse to answer any questions, and that you may withdraw from the study at anytime.
TO: Yes
Enciliah: How would you describe software quality management at your company?
TO: It's systematic and manual process of determining whether a product or service meets specified requirements of a use case and specification.
Enciliah: What rules and regulations are in the quality management process?
TO: To get requirements spec out as clear as possible > Get Prioritized and into Development > Testing to make sure it's a working deliverable > Release to customers
Enciliah: What are some of the challenges?
TO: Requirements spec Reviews and Risk based Testing
Enciliah: How can the process be improved?
TO: Risk based Testing - Consider - FMEA: failure modes and effects analysis and Requirements spec Reviews - Perhaps making sure that all stakeholders reach a consensus and every team player is on the same ground before rushing into implementation and development, so that the client and management are sure that the developers will deliver the right thing at the end of the development cycle.
Enciliah: Are you involved in testing and what level of testing is done?
TO: Yes, very involve in system testing and acceptance testing.
Enciliah: What processes are in place to ensure customer satisfaction?
TO: Consistent communication to understand and get clear expectations.

Enciliah: That's it
TO: Is there anything else that you would like to add?
Participant: No
Enciliah: Thanks for taking the time to talk with me today. I really appreciate it

APPENDIX E: SPOTIFY ENGINEERING CULTURE

Part 1



Part 2