Modelling the critical success factors of agile software development projects in South Africa

by

Tawanda Blessing Chiyangwa

submitted in accordance with the requirements for
the degree of

Doctor of Philosophy

in the subject

Computer Science

at the

UNIVERSITY OF SOUTH AFRICA

SUPERVISOR: Professor Ernest Mnkandla

26 January 2017

DECLARATION

Name: Tawanda Blessing Chiyangwa

Student number: 4991-849-4

Degree: Doctor of Philosophy in Computer Science

I declare that **Modelling the critical success factors for agile software development projects in South Africa** is my own work and that all the sources that I have used or quoted have been indicated and acknowledged by means of complete references.

I further declare that I have not previously submitted this work, or part of it, for examination at UNISA for another qualification or at any other higher education institution.

______ ____ _17 February 2017_____

SIGNATURE DATE

(Mr T B CHIYANGWA)

ACKNOWLEDGEMENTS

I am sincerely grateful to my promoter, Professor Ernest Mnkandla for his patience, guidance, rigorous reviews and insightful feedback. My special thanks to the successive management of the College of Science Engineering and Technology for their persistent encouragement. The University of South Africa deserves special thanks for funding my studies through the Masters and Doctoral Support Programme, managed by the College of Graduate Studies under the able leadership of Professor Elana Swanepoel, who were very supportive. Special thanks also go to my wife Brenda Zenda Chiyangwa, my language editor Helen Alexander, my brothers Leonard Chiyangwa and Fungai Chiyangwa, my sister Beatrice Chiyangwa, and her husband Professor Joseph Chisasa, my father David Dzvakakuyambwa, and my mother Rugare Takawira, for providing family support; I am grateful. Finally, I extend my gratitude to my son Ethan Chiyangwa for being deprived of the much needed fatherly quality time he deserved.

DEDICATION

This thesis is dedicated to my father David Dzvakakuyambwa and my mother Rugare Dzvakakuyambwa whose encouragement and tolerance enabled me to be the Tawanda Chiyangwa of today. Dedications are also to my wife, son, relatives and supervisor.

ABSTRACT

There are several critical success factors suggested for why agile software development projects succeed, including organisational and process factors. Although there are an increasing number of identified critical success factors, IT professionals lack the modelling techniques and the theoretical framework to help them meaningfully understand their influences. To solve this problem, this study developed a model by employing the following theories: Theory of Reasoned Action (TRA), Theory of Planned Behaviour (TPB), and Unified Theory of Acceptance and Use of Technology (UTAUT) to create a fit model for agile software development projects. The research sought to answer the question: What are the critical success factors that influence the success of agile software development projects?

The literature review considers the continued failure of agile and traditional software development projects which have led to the consideration of, and dispute over, critical success factors — the aspects most vital to a methodology's success. Though TRA, TPB and UTAUT have previously been applied to agile methodologies, empirical models have not been completely integrated to develop a fit model. This research sought to fill this gap.

Data was collected in South Africa through a web-based survey using structured questionnaires and an interview guide. Face-to-face interviews were done to identify the critical success factors in agile projects. The data was captured and analysed for descriptive statistics, convergent and discriminant validity, composite and internal reliability, and correlation in order to inform the structural equation modelling (SEM). SEM was used to test the research model and hypotheses to answer the research questions.

The results informed development of a comprehensive model that could provide guidelines to the agile community. The study contributes towards understanding the critical success factors for agile projects. It examined direct, indirect and moderating effects, and the findings contribute towards developing a framework to be used by agile professionals.

These key result shows that organisational factors have a great influence on performance expectancy characteristics. To ensure success of agile projects, managers are advised to focus on the effect of the organisation's environment, culture and policies on performance and effort expectancy.

Keywords

Software quality, agile software development project, software projects, Theory of Reasoned Action, Theory of Planned Behaviour, failure factors, critical success factors, agile methodology, traditional methodology, and Unified Theory of Acceptance and Use of Technology

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AGFI	Adjusted Goodness of Fit Index
AMOS	Analysis of Moment Structures
ASD	Adaptive Software Development
AVE	Average Variance Extracted
C.R.	Composite Reliability
CEO	Chief Executive Officer
CFA	Confirmatory Factor Analysis
CFI	Comparative Fit Index
CMIN	Chi-square
CMIN/DF	Chi-square per Degrees of Freedom
CSFs	Critical Success Factors
DF	Degrees of Freedom
DOI	Diffusion of Innovation
EFA	Exploratory Factor Analysis
FDD	Feature-driven development
GFI	Goodness of Fit Index
ICT	Information and Communication Technology
IS	Information System
IT	Information Technology
JAD	Joint Application Design
KMO	Kaiser-Meyer-Olkin
ML	Maximum Likelihood
MM	Motivation Model
MPCU	Model of PC of Utilisation
NFI	Normed Fit Index
PBC	Perceived Behavioural Control
RMR	Root Mean Square
RMSEA	Root Mean Square Error of Approximation
SA	South Africa
SCT	Social Cognitive Theory
SD	Standard Deviation
SEM	Structural Equation Modelling
SPSS	Statistical Package for Social Scientists
TAM	Technology Acceptance Model
TPB	Theory of Panned Behaviour

TRA	Theory of Reasoned Action
UK	United Kingdom
UNISA	University of South Africa
UTAUT	Unified Theory of Acceptance and Use of Technology
VIF	Variance Inflation Factor
XP	Extreme Programming

GLOSSARY

Agile Software Development	Agile software development is an evolutionary, highly collaborative, disciplined, quality-focused approach to software development, whereby potentially shippable working software is produced at regular intervals for review and course correction.
Critical	Extremely important, absolutely necessary for the success of something (e.g. agile software development projects) and likely to get out of control if neglected.
Critical factors	Those factors identified as extremely important for having a greater impact on the success or failure of a system. (e.g. actual success for agile software development projects)
Critical success factors	Those aspects of an organisation that are identified as vital for a successful goal to be achieved (e.g. successful implementation of agile software development projects). They may include but are not limited to; processes, individual and organisation skills, functions, techniques, technologies and any other external influences.
Cross- sectional studies	A survey method where data from the study population is gathered at a single point in time. The needed observations, measurements or comparisons of all targeted population, or representative subsets, are taken at one specific point in time. Most likely over a period of days or weeks.
Dependent Variable	A variable of primary interest to the study, also known as the criterion variable.
Dichotomy	This refers to a division into two parts within a thinking, belief, philosophy or paradigm. A research dichotomy refers to the different contradicting or opposing paradigms or approaches in the research.
Eigen value	Eigen value or character value is a factor by which the independent eigenvector or character vector is scaled when multiplied. The eigenvectors on the other hand are the non-zero vectors that, after being multiplied by the matrix, remain parallel to the original vector.
Framework	A general overview or outline of a group of interlinked components or items that supports a particular approach to achieve a consistent objective. It helps to appreciate the ideas that are already established to do something in a similar way or use a given method. A theoretical framework of the study is a structure that can hold or support a theory of the research work and serves as a basis for conducting a particular research.

	A conceptual framework on the other hand is the operationalisation of the theory. It expresses the researcher’s own position on the problem and gives direction to the study.
Independent Variable	A variable that influences the dependent or criterion variable and accounts for (or explains) its variance.
Moderating variable	The moderator or the moderating variable is one that has a strong contingent effect on the independent variable and dependent variable relationship. That is, the presence of a third variable (the moderating variable) modifies the original relationship between the independent and the dependent variables.
Pre-testing	A trial run with a group of respondents for the purpose of detecting problems in the questionnaire instructions or design, and seeing whether the respondents have any difficulty understanding the questionnaire or whether there are any ambiguous or biased questions.
Questionnaire	A pre-formulated written set of questions which respondents answer, usually within rather closely defined alternatives.
Reliability	The extent to which research findings would be the same if the research were to be repeated at a later date, or with a different sample of subjects.
Sample	A sample is a subset of the population, comprising some members selected from the population.
Validity	The extent to which the data collected truly reflects the phenomenon being studied.

PUBLICATIONS FROM THE STUDY

Papers under review

Chiyangwa, T., & Mnkandla, E. (2017). Modelling the critical success factors for agile software development projects in South Africa. *South African Journal of Information Management*.

CHAPTER 1: INTRODUCTION

This chapter describes the motivation, the research background and the purpose of this study. The existing research on agile and traditional methodologies practices is explained and the significance of the critical success factors of agile software development projects are discussed. Lastly, the research problem, the research questions, the research objectives, the delineations and the layout of the study are explained.

1.1 Motivation of the research study

It has long been understood that a large portion of software development projects fail. Recent estimates note that approximately 65% of software development projects in general do not meet their time frame and incur high costs to the organisation (Bossini & Fernández, 2013). Organisations are spending a lot of money adopting agile development processes with an expectation that the software projects are going to be completed on time with less cost (Ambler, 2009; Chow & Cao, 2008). However, many software development projects, even those adopting an agile methodology, continue to incur costs to the company which were not budgeted for, and are completed beyond the initially agreed timeframe (Dyba & Dingsøyr, 2015).

The selection of inappropriate methodologies to manage software development projects contributes to the failure of these projects (Bossini & Fernández, 2013; Stankovic, Nikolic, Djordjevic & Cao, 2013). Currently, there is confusion over which software development process to choose in different circumstances. Consequently, there is a need for software development managers to understand when it is appropriate to use agile and when to use traditional methodologies (Taromirad & Ramsin, 2008).

Although a number of software development methodologies exists, each with related critical success factors, software project managers find it difficult to select the most appropriate one (Nguyen, 2016). This is partially because software development professionals tend to be passionately devoted to and subjective towards the software engineering methodologies in which they have lot of experience (Bossini & Fernández, 2013).

When the CSFs of agile software development projects are not logically connected with project objectives and their specific environment, this leads to failure to deliver software projects on time, and at a higher cost to the software project than budgeted for by the organisations (Chow & Cao, 2008; Misra, Kumar & Kumar, 2009). There is no complete research framework to identify and develop insights into all relevant critical success factors for agile development projects and their linked constructs which have a vital meaning to the software development community and organisations (Dyba & Dingsøyr, 2015; Misra, et al., 2009).

According to Stankovic et al. (2013), there are several critical success factors (such as top management support and training, co-location of the whole team, and facilities with proper agile-style work environments) which are required for agile software development projects to be successful. In addition, a successful project requires dedicated people and sponsors who are directly involved in the software project and people who are enthusiastic about putting everything in practice, confronting resistance, and encouraging the management and employees towards the organisation’s mission and vision through communication (Cohn & Ford, 2003).

Furthermore, agile projects succeed in an organisational culture that is agile-friendly. Culture defines the core beliefs, values, norms and social customs of an organisation (Rogers, 1983). This implies that Agile Software Development Projects succeeds in a culture where individuals feel contented and empowered by having liberty (Boehm & Turner, 2003c). In addition, the agile approach succeeds in a cooperative culture, as opposed to a hierarchal culture (Reifer, Maurer & Erdogmus, 2003).

Moreover, agile software projects are fruitful when the agile methodologies are correctly used within the organisation by agile professionals (Lindvall, Basili, Boehm, Costa, Dangle, Shull, Tesoriero, Laurie Williams & Zelkowitz, 2002). Lastly, organisations that are dynamic and fast changing tend to find agile software development methodologies appropriate (Abrahamsson, Salo, Ronkainen & Warsta, 2002).

Despite agile professionals having identified several critical success factors, agile software projects still fail (Chow & Cao, 2008). Marnewick (2012) discovered that, 12% of software projects failed in 2011, which means that approximate R14 billion was wasted in 2011 (as shown in Figure 1.1). Joseph (2013) forecast that the increase in over-spending in software projects will increase to approximately R150 billion in 2016, which means that more than R18 billion could also be wasted in 2016 in South Africa.

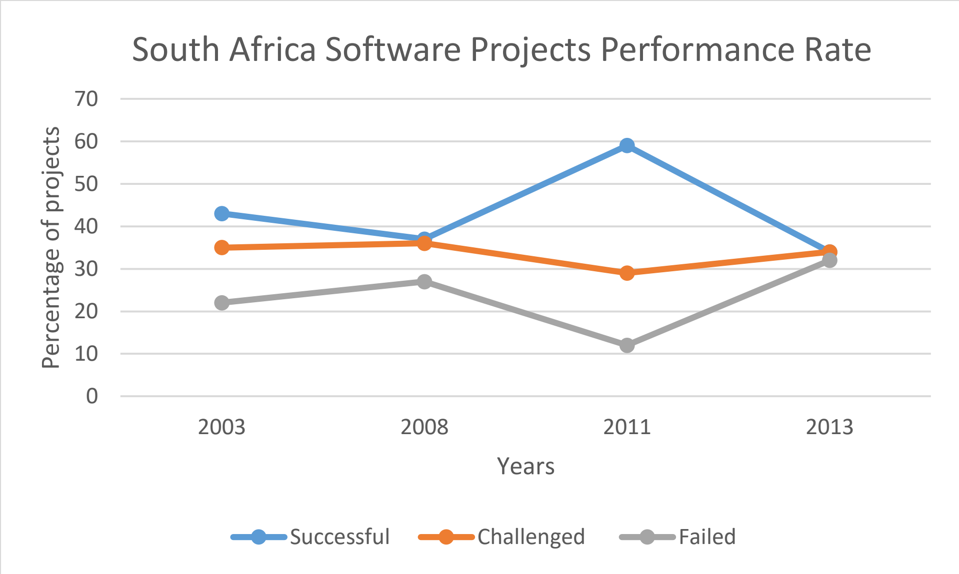


Figure 1.1: Source: Joseph (2013)

In light of the above, there is a need to model the critical factors that influence the success of software projects that use agile methodologies in order to avoid a waste of money, time and organisational resources. Unfortunately, not all the critical success factors that affect agile software projects can be addressed at once, however modelling the complexities of software development using new probabilistic techniques presents a positive way forward.

1.2 Background of the research study

Despite the potential benefits of using agile methodologies, software development professionals have been slow in adopting them. In a global study, Chan and Thong (2009) discovered that: 60% of companies surveyed were neither using agile nor any traditional methodologies, only 6% followed a methodology rigorously, and 79% of those did not use any methodology and did not intend to adopt one. One of the reasons for the lack of acceptance of agile specifically is that early adopters of technology are highly resistant to changes to new technology. Further, agile methodologies are assumed to be universally applicable and people assume that they can be adjusted to software development during a project. The acceptance of agile software development methodologies remains a persistent challenge that attracts agile professionals' attention (Joseph, 2013).

Joseph (2013) explained that more than 65% of software projects in South Africa were perceived as being failures or facing challenges in 2013 (as shown in Figure 1.1). This might have been because software development projects were not completed in time or within the estimated budget. Joseph (2013) further explained that only 34% of the software development projects were perceived as successful (as shown in Figure 1.1). In a similar study, Marnewick (2012) found that 59% of the South African software development projects surveyed in 2011 were perceived as being successful and 41% of the software projects were perceived as unsuccessful or facing challenges (as shown in Figure 1.1).

The Industrial Development Corporation (2012) predicts an increase in software project spending in South Africa of R154 billion in 2016. This implies that R18.48 billion could be wasted in 2016 based on the 2011 failure rate shown by Marnewick (2012). This amount could potentially increase to a staggering R49.28 billion if the current 32% failure rate is considered (as shown in Figure 1.1) (Joseph, 2013). Joseph (2013) showed that approximately R30.8 billion could be lost in failed software projects. This raises the concern whether all software development project managers are professionals who know what they are doing.

All software development project success depends on several factors such as having a software project manager with an effective management style and decision making. The main building blocks for the success of an agile project specifically are time, acceptance and understanding of agile methodologies, quality, budget and scope. On the other hand, there are several factors which agile management and agile professionals fail to consider even though they have experience from previous agile software projects namely; people factors, organisational factors, culture factors, process factors, project factors, political factors and technical failure factors (Chow & Cao, 2008; Misra et al., 2009; Dyba & Dingsøyr, 2009; Koch, 2005; Lindvall, Muthig, Dagnino, Wallin, Stupperich, Kiefer & Kahkonen, 2004; Cohn & Ford, 2003).

Mixed research study was used in the current study to understand the practical experience of agile professionals with agile project success, exploring the human, project, process, organisational, technical, political, and technological factors in South Africa organisations. Government in South Africa and agile organisation have created a conducive environmental for creation and growth or transformation of enterprises. This consists of good communication, good accessibility of resources, project preparation, budget allocation, modification control process of software projects, just to name a few (Joseph, 2013; Chow & Cao, 2008; Misra et al., 2009; Dyba and Dingsøyr, 2009). However, there is limited information available pertaining to the critical success factors that have an effect on agile software development projects (Joseph, 2013).

The scope of the current study is solely within South Africa. This study has implications for positive social change in organisations as they will be better able to understand the critical success factors of agile software development projects. This study will also enable organisations to develop strategies to improve agile software development projects and cost benefits leading to higher profitability and productivity of software projects.

There has been argument in the literature about agile software development projects from the beginning (Koch, 2005). The critical success and failure factors of agile have been the subject of major debate for over a decade (Dyba & Dingsoyr, 2009; Cohn & Ford, 2003; Koch, 2005). Some of the critical success factors mentioned by Misra et al. (2009), Chow and Cao (2008), Nerur, Mahapatra and Mangalaraj (2005), Chan and Thong (2009), Joseph (2016), Dyba and Dingsoyr (2009), Cohn and Ford (2003), Lindvall et al. (2004), Koch (2005), Venkatesh et al. (2003), Highsmith (2002a), and Nguyen (2016) are shown in Table 1.1 below.

Table 1.1: Success factors of the agile projects (Marnewick & Labuschagne, 2009; Augustine, Sencindiver & Woodcock, 2005; Misra et al., 2009; Ceschi, Sillitti, Succi & De Panfilis, 2005)

Dimension	Factors
Organisational	Facility with proper agile-style work environment
	Collocation of the whole team
	Cooperative organisational culture instead of hierarchal
	Oral culture placing high value on fluid, face-to-face communication
	Agile methodology is universally accepted in the organisation
	Reward system that is for appropriate for agile
	Committed sponsor or manager
	Good management or executive support
People	Good customer relationship
	Coherent, self-organising teamwork and motivated team
	Managers who have light-touch or adaptive management style
	Team members with high competence and expertise
	Team members with great motivation
	Managers knowledgeable in agile process
Process	Strong communication focus with daily face-to-face meetings
	Following agile-oriented requirement management processes
	Project scope is well-defined
	Agile professionals follow an agile-friendly progress tracking mechanism
	Following agile-oriented configuration management process
	Following agile-oriented project management process
	Strong customer commitment and presence

Dimension	Factors
Technical	Project imposes a well-defined coding standard up front
	Project pursues simple design
	Project pursues vigorous refactoring activities
	Project maintains the right amount of documentation
	Correct integration testing
	Delivering the most important features first
	Appropriate technical training to team members
Project	Project with a dynamic, accelerated schedule
	Project with no multiple, independent teams working together
	Projects with up-front cost evaluation completed
	Project type being of variable scope with emergent requirements
Cultural	Organisation does not have a bureaucratic management structure
	Organisational culture is customer-centric
	Management culture supporting the decisions of the agile professionals
	Organisation encourages rapid feedback from customers as culture
	Organisation has the culture of trusting other agile professionals
Social	Organisation expert use encourages use of agile methodologies
	Organisation team members motivate use of agile methodologies
	Management encourages use of agile methods
	Management from different organisation motivates use of agile methodologies
	Friends from different organisation encourage use of agile methodologies
	Agile methodology use is expected
	Majority of people valued by individual use agile methodologies
	Community approves of individual's use of agile methodologies
Success	Quality (delivering project outcome)
	Scope (meeting all requirements)
	Time (delivering on time)
	Cost (delivering within estimated cost)
	Quality (delivering good project outcome)
	Scope (meeting all objectives)
	Cost (delivering within estimated effort)
Political	Governmental policies influence the organisation
	Management policies influence the organisation

Dimension	Factors
	International policies influence the organisation
	Local politics influence the organisation

Existing research has attempted to examine the usage and acceptance of agile methodologies from the use and technology adoption perspective (Venkatesh et al., 2003; Koch,2005). These studies treat agile methodologies as usage and technology innovations, and make use of the usage and technology adoption models, such as TAM, TRA, DOI, and UTAUT (Venkatesh et al., 2003; Chan & Thong, 2009). Although these models have been found to be suitable for examining the individual's perception of agile software development projects, they focus mainly on the usage and technology characteristics, which can be categorised into perceived usefulness and perceived ease of use.

Many theories have been used to predict users' usage and acceptance of agile methodologies from the use and technology adoption perspective (Venkatesh et al., 2003; Chan & Thong, 2009). One school of thought starts with the Theory of Reasoned Action (TRA), followed by the Theory of Planned Behaviour (TPB), the Technology Acceptance Model (TAM) and the Unified Theory of Acceptance and Use of Technology (UTAUT); and the process of model building continues. All of the usage, acceptance, and intention based theories and models in this set look at the cognitive processes of individuals even though the analysis then aggregates the individual results in order to try to predict how an individual, potential end user will act regarding technology adoption. Therefore, UTAUT have not been used in the agile software development project as the recent modified model.

Non-technology factors have been largely neglected in previous research. Individual, political, and cultural factors, performance expectancy, effort expectancy and organisational characteristics can also play critical roles in the usage and acceptance of agile methodologies.

Further, there is lack of empirical research in the domain of agile software development projects, with research mostly confined to qualitative case studies. Johansson (2000) explains the inappropriateness of using statistics in this area because of difficulties linked with interpreting the results and many existing uncertainty factors in software projects. On the other hand, Dyba and Dingsøyr (2015), Misra, et al. (2009) and Chow and Cao (2008) explained the mathematical background of their research by formulating multiple regression models in their studies. This research seeks to fill this gap in previous research and solves the research problem by modelling the critical success factors of agile software development projects as a way forward.

There is an urgent need to conduct a critical review of the existing literature to develop a conceptual framework for agile methodology usage and acceptance in organisations in South Africa (The study is limited to one country (South Africa) because contextual factors, such as culture and politics will vary enormously from country to country). In developing this conceptual framework, it will be important to consider multiple aspects – including individual, political, culture, organisational, and technical characteristics – in order to arrive at a balanced and comprehensive understanding of agile methodology usage and acceptance in successful agile projects.

Hansson, Dittrich and Zarnak (2006) investigated the differences between agile software development practices in several organisations. In a cross-sectional survey study, Chow and Cao (2008) showed that only 10 out of 48 hypotheses were critical to success of agile software development projects in relation to industrial practices. They have therefore started to identify and show insight into the critical success factors for agile software development projects. Both sets of findings are in agreement with some of the 12 principles of agile practices laid down in the Agile Manifesto practices as well.

1.3 Research problem

Software development projects adopting agile methodologies face a number of challenges which have emerged through various studies which found that:

- Many agile professionals were not equipped with the necessary skills (Chow & Cao, 2008).
- Most agile professionals studied lacked training (Misra et al., 2009), with outsourced training sometimes provided by private consultants who themselves had insufficient knowledge and practical training experience (Misra et al., 2009).
- The agile methodology was designed to resolve concerns in traditional software development projects but these issues continued to be problematic (Misra, Kumar & Kumar, 2006).
- Although many agile professionals and managers received training on agile, they still struggled to appreciate the value thereof and some of them preferred traditional software development approaches (Chow & Cao, 2008).

Recently, researchers have cited the key problems in implementing agile software development projects as being that the relevant software development professionals have not been effectively trained to handle the changes, while the software project experts do not adequately understand or explain the main difference between traditional and agile software development projects (Chow & Cao, 2008).

Many organisations are still confused about which methodology should be used to develop software projects. There exist several software development methodologies each with related critical success factors, and software development professionals and their management find it difficult to select the most appropriate methodology (Nguyen, 2016). This is exacerbated by the fact that agile professionals tend to be passionately devoted to agile and will advocate for its use without considering organisational and other factors (Bossini & Fernández, 2013).

Chow and Cao (2008) have confirmed the importance of using a modelling technique to identify the critical success factors of agile software development projects. There remains a gap in literature that this research aims to fill through exploring organisations' views and opinions concerning agile software development projects in South Africa and how they perceive the critical success factors in their organisation. This calls for the development of a theoretical framework that exhaustively determines the critical success factors that influence the implementation of software development projects using agile methodologies.

1.4 Research questions

The following research questions were developed in response to the research problems stated above. These questions then guided the current study.

1.4.1 Primary research question

What are the critical success factors that influence the success of software development projects using agile methodologies?

1.4.2 Secondary research questions

1. How do agile professionals perceive the adoption of agile software development projects in South Africa?
2. What is the most appropriate theoretical framework which can be adapted to model the critical success factors of agile software development projects?
3. How can critical success factors be structured into a framework that can inform agile professionals and the community?

1.5 Research purpose

The purpose of this study is to develop a research model to evaluate the critical factors that influence the success of agile software development projects which can inform the agile community, professionals and management about the success of agile software development projects in South Africa.

1.6 Objectives

Using South Africa as a context of analysis, the study's objectives are listed below.

NB: An expert in a method of project management activities and is engaged in a specified activity as one's main paid occupation rather than as a pastime.

1.6.1 Primary Objective

The primary objective of this study is to identify and provide insight into the critical success factors that influence the success of software development projects using agile methodologies. To accomplish this, the following specific objectives were pursued:

1.6.2 Secondary objectives

1. To determine how agile software development project success is perceived and evaluated within organisations in South Africa;
2. To determine the most appropriate theoretical framework which can be adapted to model the critical success factors; and
3. To construct a structural equation model for the critical success factors of agile software development projects.

1.7 Delineations

The following delineations are applicable:

1. The study did not examine the critical success of agile software development projects globally.
2. The study did not examine any other population sample except the Agile Alliance and its user groups (agile professionals) in South Africa.

1.8 Layout of the research study

The rest of this study comprises the following chapters:

Chapter 2: Literature review: This chapter outlines agile project failure and success in South Africa. Traditional and agile methodologies are examined and the critical success and failure factors of agile software development projects are reviewed.

Chapter 3: Theoretical framework: This chapter examines theoretical models for agile software development projects. It further reviews the theories of traditional software development projects and attempts to link them to available empirical evidence.

Chapter 4: Research design and methodology: In this chapter the empirical design for the study is articulated. The research methods used in the study and the survey approach are discussed. Further, the data collection instruments and the methods of analysis are set out. Descriptions of the research design are outlined, giving the respective merits and disadvantages.

Chapter 5: Data analysis: A discussion of how the survey data were analysed and interpreted is presented in this chapter. The chapter starts with a presentation of the descriptive and inferential statistics and concludes with more robust structural equation modelling techniques.

Chapter 6: The effects of moderators: In this chapter, the effects of the moderating factors on the research model is explained. This chapter also evaluates the hypotheses of this study.

Chapter 7: Discussion of results: In this chapter, the results from the analysis of the primary data are synthesised in order to get a clear understanding of the relationship between the various critical success factors of agile software development projects.

Chapter 8: Evaluation of the research and conclusion: In this chapter, the conclusions of the study are presented and recommendations are made for further research.

CHAPTER 2: LITERATURE REVIEW

A preliminary literature survey was conducted to establish the validity of the research and to set out the foundation for the remainder of the study. This chapter provides a review of the literature on the trends and use of software development methodologies, and provides an insight into agile methodologies. The first section discusses traditional and agile methodologies. The next section explains the failure factors of agile software projects, such as organisational, people, process and technical factors. Lastly, there is a discussion of the success factors of agile methodologies (including organisational, individual, process, social, political, vendor, cultural, project, and technical factors) and some thoughts are presented on how this research can help overcome some limitations in the existing body of knowledge.

2.1 Traditional methodologies

Software projects emerged as early as the 1950's, with various processing methods adopted but there is still no proven golden standard. There is a large spectrum of processes, from unstructured analysis to ill-defined analysis, and from the plan-driven approaches based on the capability maturity model to the modern goal-oriented software engineering methodology. However, in terms of distinctive characteristics, traditional software development methods can be characterised as follows: the prototyping model, the waterfall model, the spiral model and the code-and-fix model. The section below on traditional methodologies explains the most common types, linking them to critical success factors to help illustrate the differences between each method.

2.1.1 The Prototyping Model

Humphrey (1989) refers to prototyping models as the methodologies that intend to diminish the obligation of doubts through the protests of plans of organisation behaviour. Weinberg (1991) defines prototyping models as the prototyping life cycle, where a software programmer utilises an extraordinary determination prototyping tool to produce the most protruding chunks of a program and then working with customers to improve it until the prototype is sufficiently worthy (as shown in Figure 2.1). Therefore, the software is either created or rejected as a new software system. Furthermore, McConnell (1996) defines prototyping models as evolutionary prototyping, which focuses on the most favourable situations for the model, for instance when neither the software programmers nor the users understand the end user interface or the final design well. The main disadvantage of this technique is that it is difficult to recognise at the beginning of the project how long it will take to build a tolerable software system. Moreover, Wetherbe (1991) distinguishes the relationship between the Joint Application Design (JAD) and the prototyping model technique, which includes customers and software programmers who collaborate to decide matters in prototyping models. Glass (2002) notes that JAD and the prototyping model continue to be implemented in numerous applications, particularly in situations where requirements are ill-defined. The prototyping models are regarded as predecessors to agile methods because of the iterative aspect of prototyping and the close collaboration with the clients.

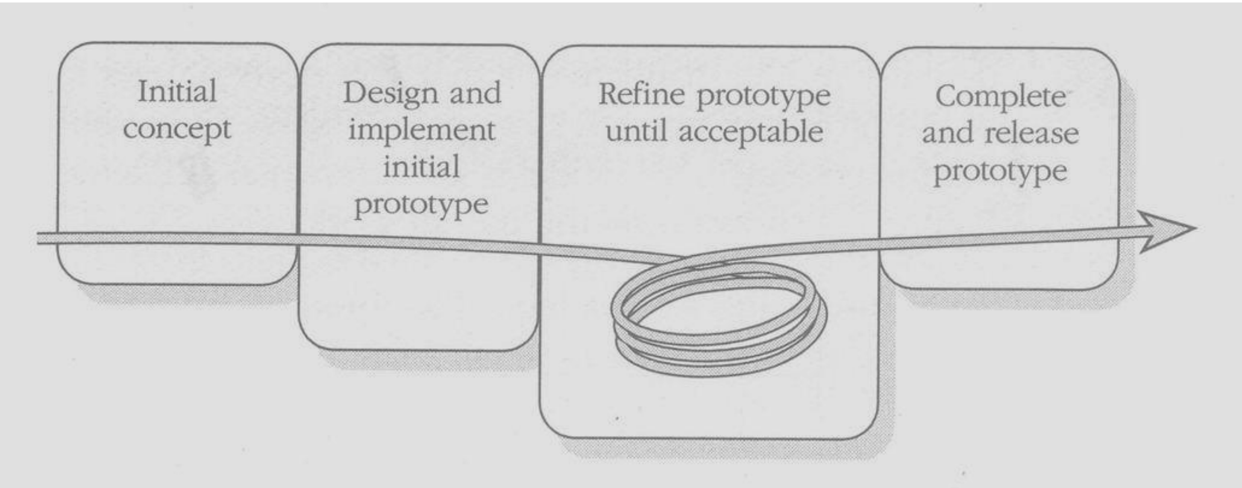


Figure 2.1: Prototype model (McConnell & Root, 1996)

The prototype model supports good communication between customer and software developer during the software development process since it allows the refinement of the prototype until an acceptable prototype is made. Therefore, organisational success factors are critical success factors for the prototype model supports because of the reliance on good communication between stakeholders and software developers.

2.1.2 The Waterfall Model

The foundations of the waterfall model were initiated by Benington in 1956, basically derived from mechanical engineering (McConnell & Root, 1996). The waterfall model was proposed by Royce (1970). Although he did not mention the name "waterfall model", he used it as an example and it was established by Boehm, Brown and Lipow (1976) as one of the early models in the business. The waterfall model is still the best known and most widely used software development process, especially in the government sector (Boehm, 1988; Munassar & Govardhan, 2010). Though it is claimed to have many difficulties, it helps as the basis for other, more operative life-cycle models. The waterfall model leads project growth through an arranged order of stages from the early software concept through system testing (Munassar & Govardhan, 2010; Pichler, 2010).

Williams (1984) describes the waterfall model as manuscript-driven and notes that it works well for product cycles in which product requests are well-defined, and methodologies and technical tools are well-understood. The waterfall model helps future planning to be done but does not deliver a concrete outcome in the form of complete software until the end of the life cycle of the project (Munassar & Govardhan, 2010). The waterfall model operates well when quality requests, control and schedule and cost requests are needed (Munassar & Govardhan, 2010; Pichler, 2010).

The waterfall model operates well even when some team members are technically weak or unskilled because it delivers the project with a structure that diminishes futile struggle (Williams, 1984). The faults of the waterfall models arise from the difficulty in completely stipulating requests at the start of the project, before any design task has been completed and before any program has been transcribed. Therefore, this model is obstinate.

The waterfall model has several disadvantages: it is ill-suited for a rapid-development project and does not include the practice or presence of prototypes, which can certainly play a significant part in software projects (Boehm & Ross, 1989).

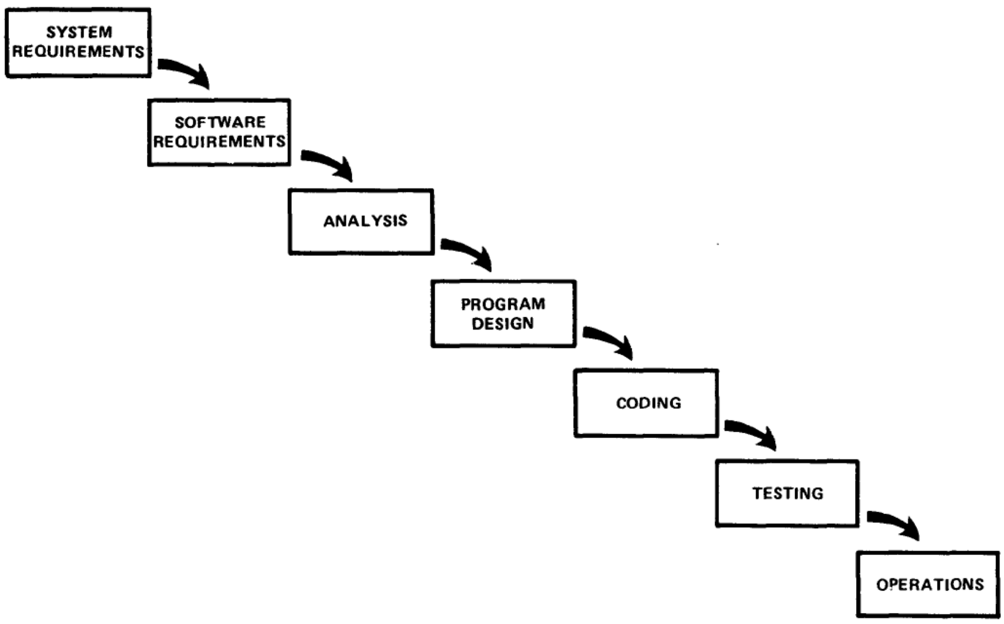


Figure 2.2: The typical process and life cycle for waterfall model (McConnell & Root, 1996; Munassar & Govardhan, 2010)

The waterfall model (as illustrated in Figure 2.2. above) includes the six phases presented below (Munassar & Govardhan, 2010):

1. System requirements: This establishes the components for building the system, including the hardware requirements, software tools and other necessary components. Examples include decisions on hardware, such as a plug-in board (number of channels, acquisition speed, etc.), and decisions on external pieces of software, such as databases or libraries. (Pichler, 2010),

2. Software requirements: This establish the expectations for software functionality and identifies the system requirements of the software project. Software requirements also focus on requirements analysis, which includes determining what interaction is needed by other applications and databases, performance requirements, user interface requirements, etc. (Pillai, Phase & Phase, 1996),

3. Analysis: This determines how users, developers, and IT managers assess the scope and feasibility of the proposed project. In addition, analysis produces a brief description of the proposed new system, a benefit or cost analysis, and preliminary schedules and budgets. (Weinberg, 1991),

4. Architectural design**:** This determines the software framework of the system to meet the specific requirements. In addition, the design defines the major components and the interaction of components, but it does not define the structure of each component (Weinberg 1991).The external interfaces and tools used in the project can be determined by the designer in the architectural design. In addition, there is detailed design which examines the software components defined in the architectural design stage and produces a specification for how each component is implemented. (Weinberg, 1991),

5. Coding: This implements the detailed design specification.

6. Testing: This determines whether the software meets the specified requirements and finds any errors present in the code.

7. Operation: This addresses problems and enhancement requests after the software releases.

In conclusion, the waterfall model supports good communication between customers, IT management and software developers during software development since it does not allow any changes to be made to the software project before completion.

These inadequacies of the waterfall model have led to the design of the spiral model.

2.1.3 The Spiral Model

The spiral model is a high risk, life-cycle model that breaks down a software project into small projects as shown in Figure 2.3 (Boehm, 1988; McConnell, 1996). Every small project statement has at least a situation or risk which needs to be resolved in regards to progressive iterations (McConnell & Root, 1996). The idea of risk is largely well-defined in this perspective, and it can mean poorly agreed requests, poorly agreed architecture, problems in the underlying technology, potential performance problems, and so on, depending on the experience of the software developers (McConnell & Root, 1996). The risk-driven nature of the spiral model is more adaptable to the full range of software project situations than are the primary document-driven approaches (such as the waterfall model) or the primary code-driven approaches (such as evolution model). After the main risks have all been dealt with, the spiral model ends as a waterfall life-cycle model which needs to terminate. Therefore, the spiral model is a risk-driven approach which is different from primarily document-driven or code-driven processes and incorporates many of the strengths of other models and resolves many of their pitfalls.

The simple idea behind the diagram in Figure 2.3 is that the project initiates on a small scale. The project increases as its iterations increases, which leads to higher risks. High risk requires decisions to be taken with regards to handling the risks (Boehm, 1988; Munassar & Govardhan, 2010). In addition, each iteration includes the six phases that are presented in bold in Figure 2.3 namely: (1) determine objectives, alternatives, and constraints, (2) identify and resolve risks, (3) evaluate alternatives, (4) develop the deliverables for the iteration and verify that they are correct, (5) plan the next iteration, and (6) commit to an approach for the next iteration (if you decide to have one).

The angular component represents progress made in completing each cycle of the spiral, and the radius of the spiral represents cost incurred in accomplishing the steps to date in the spiral model (Boehm, 1988; Munassar and Govardhan, 2010).

Some of the advantages of the spiral model as suggested by Boehm (1988) are that it: (1) focuses early attention on options involving the reuse of existing software, (2) focuses on eliminating errors and unattractive alternatives early, (3) provides a viable framework for intergrated hardware and software system development, (4) accommodates preparation for life cycle evolution, growth and changes of the software product, and (5) provides a menchanism for incorporating software quality objectives into software product development.

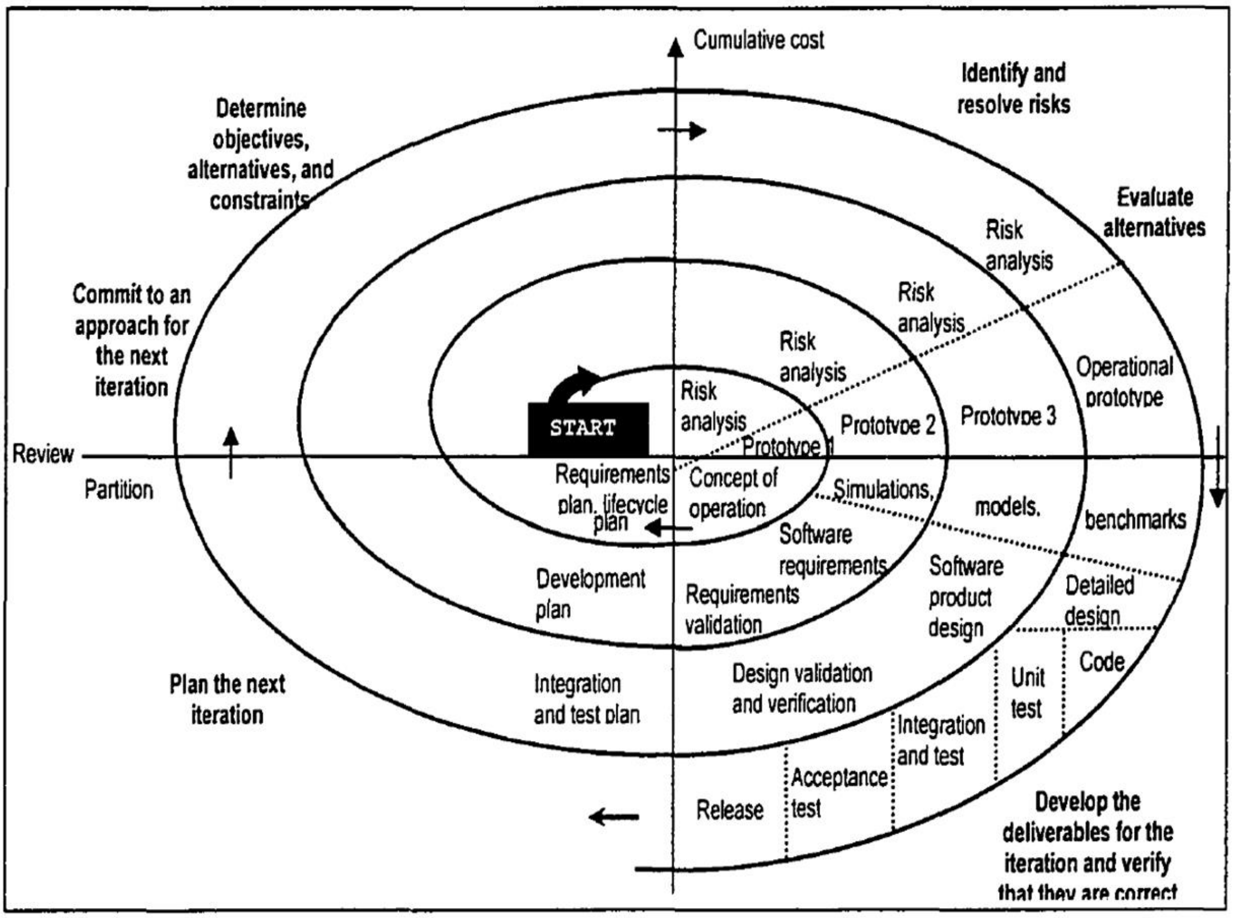


Figure 2.3: The typical process and life cycle for spiral model (Boehm, 1988; McConnell, 1996)

One of the major drawbacks of the spiral model is that it is complicated (Munassar & Govardhan, 2010) and needs careful, attentive and well-informed management. In addition, it may be challenging to define objectives, evaluate alternatives and set verifiable milestones that show whether the project is ready to go to the next phase (Munassar & Govardhan, 2010). The spiral model relies on risk-assessment expertise and need for further elaboration of spiral model steps is required (Boehm, 1988).

In conclusion, the spiral model can be costly and has a high risk of failing if you do not have enough experience. As a result, technical, organisational, people and process critical success factors are not taken into account and provided for in most cases.

2.1.4 The Code-and-Fix Model

Boehm and Ross (1989) explained the code-and-fix model as a model that is rarely valuable but is, however, common and among the recent models in the large organisations. The code-and-fix model is informal and consists of a few cycles of requirements specification in which the coding is tested and corrected until a final product is created. Therefore, the model is commonly implemented because it is fairly simple (as shown in Figure 2.4). There are two core benefits of using the code-and-fix model which are: (1) the code-and-fix model has no overhead in terms of time consumed in documentation, planning, code inspections, quality assurance, standard reinforcement or any other activities other than pure coding and, (2) the code-and-fix model needs very little training – one can allocate a software designer to start writing code, rather than complying with various procedures or standards (Boehm, 1988).

The code-and-fix model can only function on small proof of concept projects and its disadvantages including difficulty evaluating development, identifying risks or assuring quality. In addition, the code-and-fix model has drawbacks such as there being no design phase, and after a number of fixes, the code become so unstructured that it is very difficult to maintain (Boehm, 1988). Furthermore, with no testing preparation, the code is expensive to fix (McConnell, 1996). Finally, the code-and-fix model does not take into account the value of documentation, an organised development cycle and reviews, which can certainly play a significant role in software projects (Boehm, 1988). These drawbacks of the code-and-fix model have supported the popularity of the waterfall model.

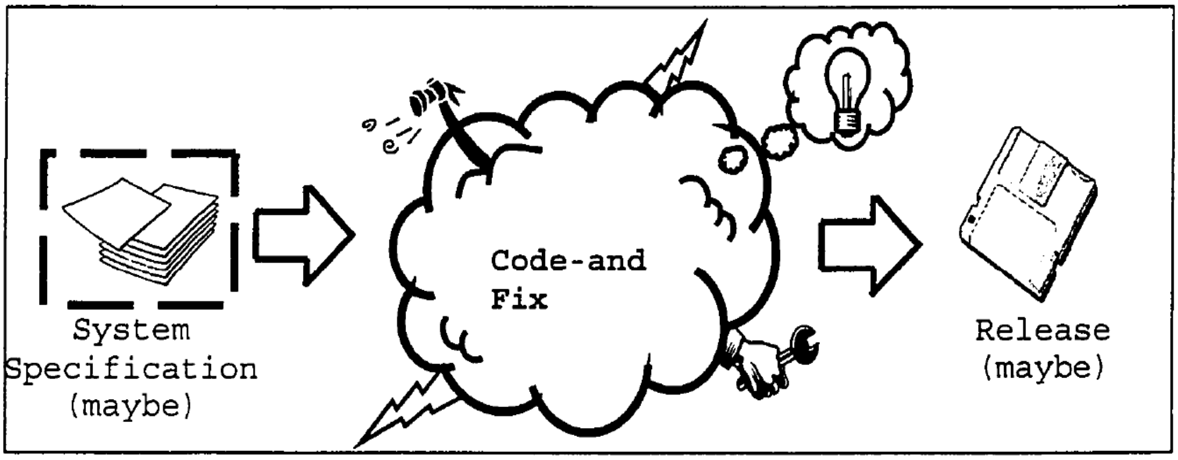


Figure 2.4: Code-and-fix model (McConnell, 1996)

In conclusion, technical and process critical success factors are generally not taken into consideration and addressed in the code-and-fix model due to the lack of a design phase during the software project process between stakeholders and software developers.

2.2 Agile methods

The Agile Manifesto was created in 2001 by a group of software development practitioners to articulate a set of 12 guiding principles that should lead to more efficient and effective software delivery (Dyba & Dingsøyr, 2008; Schwaber, 2004). The 12 principles do not offer a formal definition of agility, but are guidelines for delivering high quality software in an agile way (Dyba & Dingsøyr, 2008). Ever since the manifesto was enunciated, IT management, IT experts and researchers have been trying to explain agility and its diverse aspects (Dyba & Dingsøyr, 2008). In essence, agility involves the ability to rapidly and flexibly create and react to change in the company and technical fields. Also, it can be regarded as a light or lean methodology that encourages manoeuvrability and rapid response (Schwaber, 2004).

The principles laid out in the Agile Manifesto have led to a number of practices that are alleged to deliver better value to clients (Cordeiro, Mar, Valentin, Cruz, Patrick, Barreto & Lucena, 2008). Essential to these practices is the knowledge of self-organising teams whose members are not only co-located but also work at a pace that sustains their creativity and productivity to the company (Pichler, 2010). The principles reinforce practices that permit change in requirements at any stage of the software development process (Dyba & Dingsøyr, 2008). Additionally, clients are enthusiastically involved in the development process, enabling response and consultation that can lead to more substantial results.

Several studies on agile methods have been carried out by various researchers including investigations into the merits of agile methods, agile methods trends, extreme Programming (XP) theory, agile scaling, the internet-speed aspect of agile development, requirements concepts of agile methods, the design dimension of agile methods, the unified process of agile methods, how to introduce agile process to organisations, XP implementation, migrating agile methods, agile practice in large organisations, iterative aspects of agile methods, management challenges in implementing agile methodologies, scaling practices of agile processes, agile productivity, transition to agile methodologies from traditional methodologies, combining agile with other methods, and agile project management (Chow & Cao, 2008; Misra et al., 2009; Ambler, 2009; Cohn & Ford, 2003; Highsmith, 2002a; Karlstrom & Runeson, 2005; Reifer et al., 2003; Schatz & Abdelshafi, 2005; Schwaber, 2004; Takeuchi & Nonaka, 1986; Williams, 1984).

This study discusses the most commonly used agile methods in software development to expose the main techniques for each agile methodology.

2.2.1 Scrum

Scrum is a simple and straightforward method to manage the software development process based on the assumption that environmental and technical variables are likely to change during the process (Cordeiro et al., 2008; Dyba & Dingsøyr, 2008; Pichler, 2010; Schwaber, 2004; Schwaber & Beedle, 2002). The idea of the scrum originated in the game of rugby, referring to a strategy of getting a ball back into play (Takeuchi & Nonaka, 1986). Schwaber and Sutherland in the early 1990s improved the scrum process and managed to develop many of the early thoughts and practices for scrum when vice president of Object Technology at the Easel Corporation (Sutherland, Schwaber, Scrum, & Sutherl, 2007). By working together with employees at the Easel Corporation in 1996, the scrum process was first presented to the public at the conference of Object-Oriented Programming, Systems, Languages and Applications (OOPSLA) (Pichler, 2010; Schwaber, 1997).

Daily scrum meetings take place every working day for 15 minutes where team members exchange ideas and information (Dyba & Dingsoyr, 2009). When the scope of the sprint (repeatable work cycle) is officially determined, no extra functionality can be added but learning throughout the sprint is reflected in the product backlog. The product backlog is a set of responsibilities (dynamic queue of business and technical features) which must be carried out in the future to finish the project tasks according to the needs of the customer at the end of each sprint. Structures that deliver the most cost benefit will have a higher priority and will be developed in the succeeding sprint -- this is referred to as sprint backlog (Ahmed, Ahmad, Ehsan, Mirza & Sarwar, 2010; Pichler, 2010). Every structure is prioritised as customer's requirements change (Dyba & Dingsoyr, 2009). Towards the end of the delivery of a sprint, the software is verified to see whether it accomplishes the task as per the customer requirements (Ahmed et al., 2010). Thus software is constantly verified during the project, rather than being verified only once at the end of the software project. Finally, at the end of the sprint, a working component of the software project is sent to the customer who provides feedback to the software development team for amendments if required (Dyba & Dingsoyr, 2009; Dyba & Dingsøyr, 2015; Pichler, 2010). This is known as a sprint preview and takes a maximum of four hours (Ahmed et al., 2010).

The recommended length of each sprint is 30 days. At the sprint planning meeting clients, users, management, team members and the product owner decide on the next sprint goal to be completed. Three sprints are recommended per release (Dyba & Dingsøyr, 2015). That means there is a release of software every 90 days. Detailed understanding and face-to-face communication are constantly reinforced through daily stand-up meetings (Ahmed et al., 2010). One of the team members, known as the scrum master, is responsible for supervising the process, handling administrative and organisational tasks, and eliminating hindrances that might reduce the productivity of the team members. Scrum is most influential for teams that have eight or fewer members (Dyba & Dingsøyr, 2008).

Scrum mainly depends on organisational and management practices, in contrast to Extreme Programming (XP) which depends on actual programming practices. Scrum and XP work hand–in-hand to address various situations thus complementing each other in the agile environment.

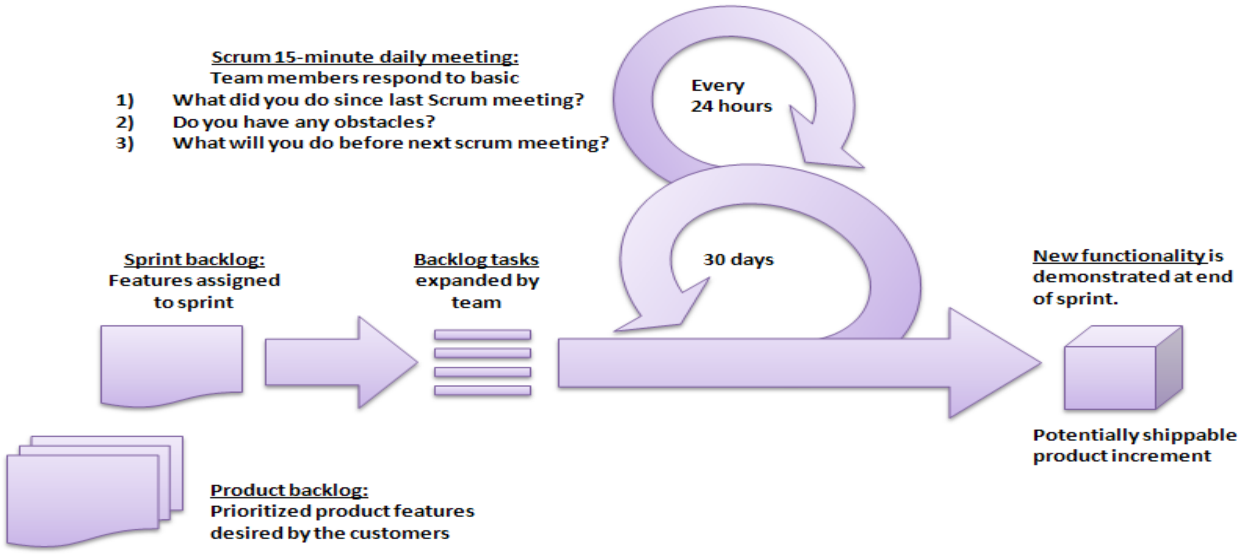


Figure 2.5: The scrum process adapted from (Boehm & Turner, 2005)

In conclusion, scrum takes into account and addresses organisational and process critical success factors which focus on organisational and management practices that support a good relationship with the customer and expertise in agile principles and processes. In addition, project and technical factors are supported, such as the focus on continuous and rigorous unit and integration testing for each and every iteration, the employment of proper platforms, technologies, and tools suitable for agility practice, a dynamic, accelerated schedule, and a small team size (7 to 10 members).

2.2.2 Extreme Programming

Extreme Programming (XP) values individuals as the most significant aspect in the development process throughout the software project (Koch, 2005; Lippert, Wolf & Roock, 2002). A customer is needed on-site as one of the main distinctive features of this method. According to Steinberg and Plamer (2003) XP emphasises the well-known agile methodologies of collaboration, early and quick development of software, and the use of essential support practices (Beck, 1999; Lippert et al., 2002). The practices of XP include: the planning game, small releases, metaphor, simple design, test first, refactoring, pair programming, collective ownership, continuous integration, on-site customer and coding standards.

XP can be defined in terms of its underlying values, and/or its practices. Lippert, Roock and Wolf (2002) defined XP as a set of values with which every member needs to be happy. These include acknowledging collective ownership (responsibility for each program module is assigned to a specific individual), feedback, respect, communication, courage and simplicity. Koch (2005) argues that these values are rather high-level and abstract, but that they build the foundation for the practices of XP. Larman (2004) suggests that practices are the main foundation He further emphasizes that practices are the core value in the implementation of real world software projects. Furthermore, Beck (1999) explained the importance of practices and values which are two separate entities. Detailed explanations of the practices, principles and values of XP are found, inter alia, in the studies by Beck (1999) and Larman (2004) and are discussed below:

- During the development process, team members must be together, preferably in one place which is referred to as "on-site team members". This guarantees easy interaction and communication among team members.

- The design of the software and its architecture must not be fixed, but should rather continuously develop throughout the development process. There should be room for change in every process done, supporting an iterative approach to the development (Larman, 2004).

- The concept of divide and conquer with regards to work done in software projects is very important. Divided tasks are called "stories". (Lippert, Roock and Wolf, 2002) Continuous integration is required – particular when a pair-programming team (an agile software development technique in which two programmers work together at one workstation) accomplishes a story – to join the stories into one before executing the code. Only if the integrated stories are 100% clean after being tested do they form part of the whole project. Alternatively, the stories or modules can be reworked by the pair programming group and can then be integrated again. Therefore, the process is done continuously every day.

- A planning meeting comprises the designer and the client's stories and metaphors, which are well-defined and assessed to ensure a simple design. Clients can select which stories the team members must work on in the next iteration, such as current incremental stories rather than future incremental which leads to less work. (Highsmith and Cockburn, 2001; Larman, 2004)

- The iteration is a fixed time interval by which every affirmed story must be carried out. (Each story is typically brief enough to be written on a 3x5-inch card.) Therefore, the end of all iterations for the project must result in a solution that is in a state to be delivered to the intended clients as per the project as requirements. (Steinberg and Palmer, 2003),

Other components which affect XP are discussed below:

1. The planning game focuses on the scope of the next operational role to deliver maximum value to the customer. Throughout the planning game, the client explains the most valued characteristics to move to the next step, thereby permitting the technical people to assess the feasibility, management, and effort of the next iteration and make sure that the implementation remains within the project scope (Lippert et al., 2002).

2. Every release in XP needs to be the smallest possible increment delivering the client's most valuable characteristics (Beck, 1999). In addition, every increment needs to be accomplished in a few weeks. XP encourages numerous methods for the actual programming, such as pair programming (where two software developers work together on one section of code), thus requiring the design to be simple, construction to be efficient and testing to be most thoroughly done. One software developer (driver) writes the code whilst the other navigates to support the driver through technical excellence and quality. A constant integration system confirms the precision of each obligation in the code repository as everything is revised continuously in real time (Steinberg & Palmer, 2003).

3. Software developers need to use computerised tests that will confirm the story before writing a single line of code for the story. Tasks on the story are disregarded until they are proven defect-free by the tests (Subramaniam & Hunt, 2006).

4. In XP, individuals are allowed to work 40 hours per week. This practice is intended to protect individual team member's productivity by ensuring they do not work overtime. Even if overtime is necessary, team members are not allowed to work overtime two weeks in a row so that they can enjoy a good quality of life (Steinberg & Palmer, 2003).

5. In XP, people work closely together, thereby taking collective ownership, which leads to a coding standard and which is essential for making the whole software project work successfully (Beck, 1999). In addition, emphasis is placed on the technical aspects of

software development, proactive response to change, use of good quality practices (such as, pair programming and test driven development), support of open and collaborative communication practices, and support of a successful software project delivery (Larman, 2004; Subramaniam & Hunt, 2006).

6. Refactoring calls for the software developers to examine their code before and after their task in order to reduce complexity. Team members meet to decide if there is a method to redesign the software project so that the final project is functioning in the most optimal and proper way. Therefore, refactoring ensures the software project does not expand from simple design to pointless complexity as more stories are added (Larman, 2004).

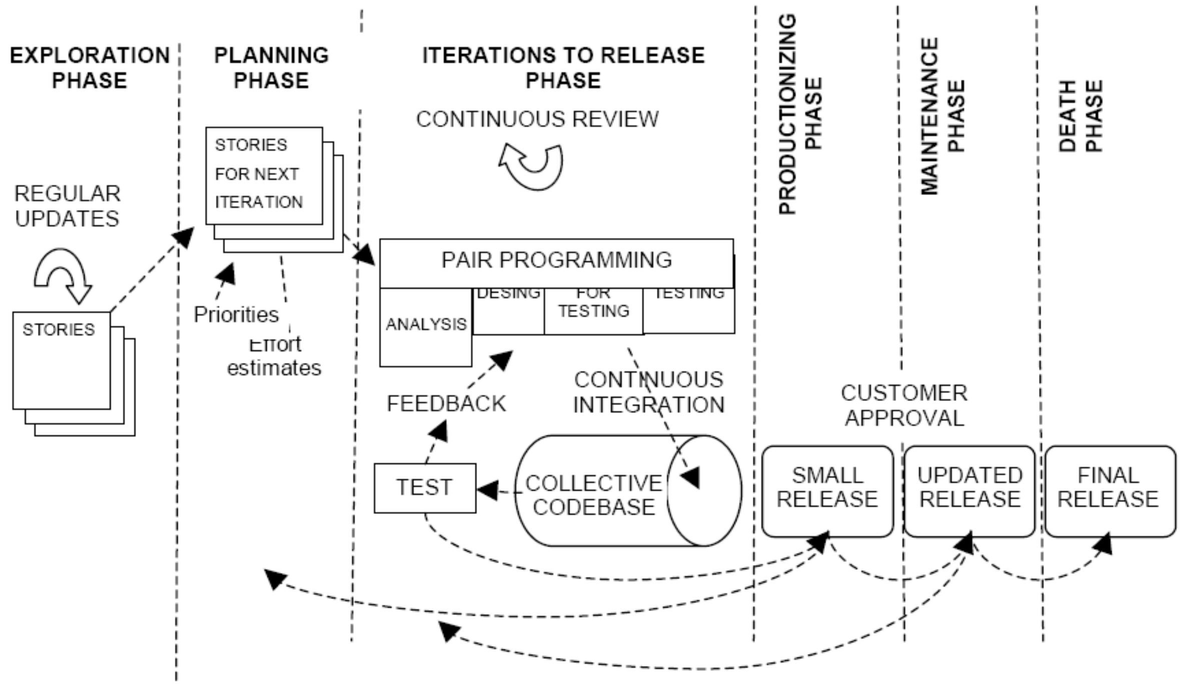


Figure 2.6: Extreme programming process (Beck, 1999)

Beck (1999) initiated Test-Driven Development and is a founding father of Extreme Programming. In Test-Driven Development the software developer writes a test before actually implementing the functionality of the particular project.

In conclusion, the values and practices of XP ensure that organisational, people and project critical success factors are considered and supported. These focus on actual programming practices that support a good relationship with the customer, sound knowledge of agile principles and processes, and project team members who work in a cohesive, self-organising manner, have great motivation and are committed. In addition, process and technical critical success factors are supported which focus on continuous and rigorous unit and integration testing for each and every iteration by employing proper platforms, technologies, and tools suitable for agile practice, a dynamic accelerated schedule, and a small team size (3 to 20 members).

2.2.3 Dynamic Systems Development Method

According to Koch (2005), the Dynamic Systems Development Method (DSDM) is an agile software development methodology which does not focus much on the prescription of programming. The Dynamic Systems Development Method originated in the United Kingdom (UK) in the mid-1990s (Koch, 2005). Moreover, The DSDM consortium is a non-profit, independent association which owns and manages the DSDM framework (Stapleton, 1997).

According to DSDM, the majority of projects that are unsuccessful due to human factors, such as lack of time and resources, rather than the scope of the technical functionality. The DSDM fixes cost, quality and time upfront, allowing the client to prioritise aspects of the functionality within this framework. To accommodate changing business requirements, DSDM ensures that every preceding phase can be reconsidered but that the current phase must be accomplished in order to move to the next phase (Stapleton, 1997).

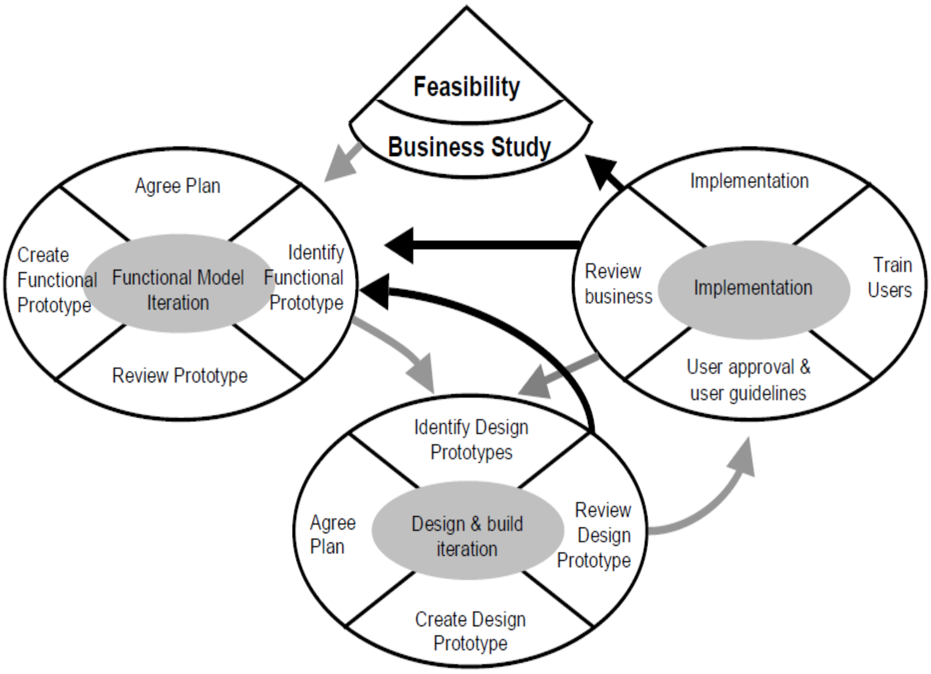


Figure 2.7: The life cycle of a DSDM project (Stapleton, 1997)

Figure 2.7 above shows the life cycle of a DSDM project consisting of five phases, namely: feasibility and business study, implementation, design and build iteration, and functional model iteration (Stapleton, 1997).

Firstly, the feasibility study stage is when the appropriate strategy for the project is determined (Stapleton, 1997). It is appropriate to select DSDM where the situation is well-defined, technical feasibility has been examined, cost has been assessed, and the period of the project is kept quite small (Koch, 2005). Furthermore, feasibility and business study requirements should be clear for the project to be successful (Koch, 2005; Subramaniam & Hunt, 2006).

Once the requirements are examined a functional model can be formed (Subramaniam & Hunt, 2006; Stapleton, 1997). Based on a list of functionality, prioritised by relevance and importance, the functional model iteration operates by prototyping and gathering functional requirements (Koch, 2005). Non-functional requirements are also stated throughout this stage. The functional model consists of functional prototypes, data models and class models (Koch, 2005). Functional model iteration refers to the initial iterative stage, which is the DSDM process where creation, identification and review of function prototypes are implemented until a plan has been agreed. (see Figure 2.7) (Koch, 2005; Subramaniam & Hunt, 2006).

This leads to the design and build iteration stage which is when the process is iterated to an adequate level to be shown to the customers (Koch, 2005). During the testing of software phase, the proposed requirements of the design and build iteration are verified and testing is completed to accomplish the requirements entirely (Stapleton, 1997). Design prototypes are identified, created, and reviewed until the agreed plan has been accomplished at which point the implementation stage commences (see Figure 2.7).

In the implementation stage, the DSDM process is moved from the development environment to a production environment. This stage comprises creating the increment review document, finishing the documentation (user approval and user guidelines), and training end users. This stage can lead back to any of the other stages (Stapleton, 1997).

DSDM includes different roles and responsibilities for team members (Stapleton, 1997). In DSDM, a programmer uses the concept of working with a consumer in a pair and this results in a strong consumer-programmer relationship (Stapleton, 1997). In addition, there are corporate duties such as management, promoter, project executive, tester, team leader and software developer. There are also other consumer duties such as advisor, ambassador and visionary (Koch, 2005).

The DSDM relies on launching, and handling the appropriate philosophy in the project (Stapleton, 1997). Team members are permitted to create decisions, to promote 100 percent commitment to the achievement of the software of project success (Koch, 2005). A person who performs well is easily recognised and compensated, and cooperation and collaboration are recognised amongst every person in the team.

The DSDM philosophies are clarified by the DSDM consortium and emphasize customer involvement (Koch, 2005). Since DSDM is a customer focused technique which comprises of active customer participation (Stapleton, 1997), customers must be carefully involved in the development and must be part of idea formulation. DSDM teams consist of both software programmers and customers, and their obligation is to be enabled to make good choices. Therefore, programmers rely on regular delivery of products to the specific customer in an approved period of time (Stapleton, 1997). This aids the team members to choose the best solution that can be attained in the specified timeframe. The best solutions are governed by agile principles. Conventionally the emphasis has been on accomplishing the planned requirements, even if they fluctuate (Koch, 2005). Incremental and iterative development allows the project to expand based on feedback from the customers. Throughout development change is possible, but changes are restricted to the present increment only. Testing is not regarded as a distinct activity but is joined in the development process of the software projects (Koch, 2005). Throughout, the project is overlooked and verified by customers, and programmers select the correct direction based on guidance from corporate. A co-operative and collaborative approach between every shareholder is essential.

A critical success factor well-supported by DSDM is that it encourages training and documentation through social interactions because team members obtain knowledge through learning from the agile expert, including being motivated by agile professionals. In addition, team members' improvement of training and documentation is influenced by different organisations that use agile methods, friends from different organisations who use agile methods, and people whom each team member values in their life who use agile methodologies.

In conclusion, DSDM supports organisational, technical and project components as critical success factor. These focus on project management emphasizing a good relationship with the customer, knowledge of agile principles and processes, and project teams of between 2 to 6 members working in a cohesive, self-organising manner, who are highly motivated and committed. However, it does not take into account important people factors, such as lack of time and resources.

2.2.4 Adaptive Software Development

Highsmith (2002a) explains Adaptive Software Development (ASD) as a complex adaptive structure which includes shareholders and team members, environments (such as technologies, societies and processes) and the outcome (which is the artefact).

The ASD life cycle consists of three parts: (1) learning, which focuses on reviewing, (2) collaborating, which focus on building and (3) speculating, which focus on planning. During these three parts, there are five steps (as shown in Figure 2.8) namely: the project initiation step which is performed once at the start of the project, in between project initiation and the final question and answer release phase, 3 steps are undertaken, namely: adaptive cycle planning, concurrent feature development and quality review (which constitute the learning loop or adaptive cycles). The final quality assurance release phase is performed once at the end of the project (Koch, 2005).

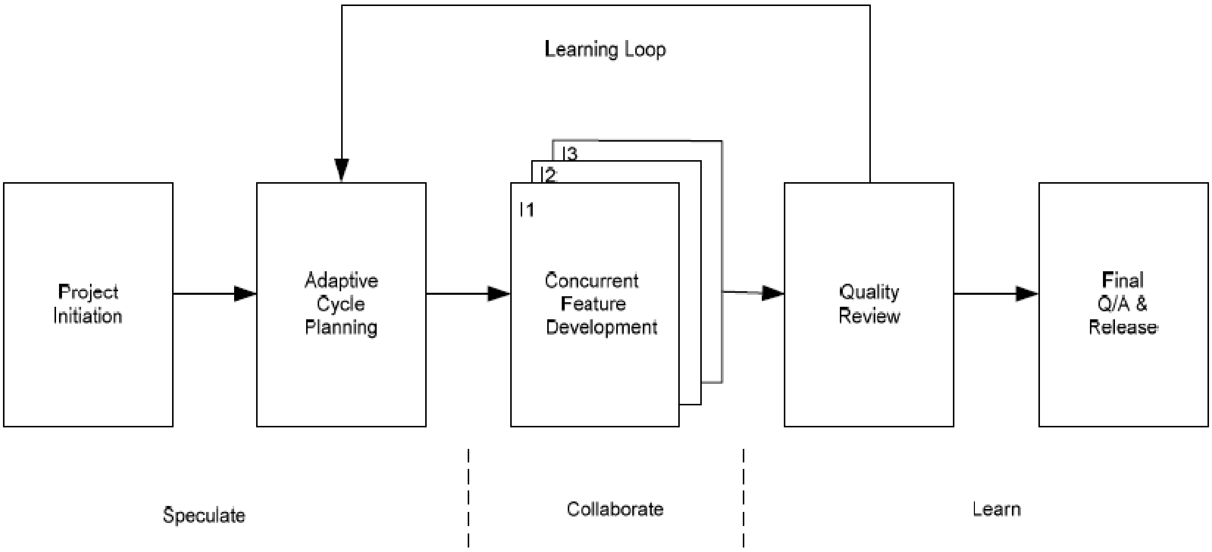


Figure 2.8: ASD life cycle model (Highsmith, 2002a)

Highsmith (2002a) notes that the life cycle of ASD is composed of six main features, in that it is

1. Mission-focused: Project requirements can be ambiguous at the starting point but the whole mission or vision statement that governs the project should be clear and firm. Hence, every adaptive cycle should move positively towards achieving the project vision (Highsmith, 2002a).

2. Feature-based: ASD emphasizes achieving outcomes rather than accomplishment of work, and the outcomes are recognised as request characteristics. Therefore, ASD is concerned with construction of characteristics (Koch, 2005).

3. Iterative: ASD goes through the learning process in a continuous loop. Several characteristics might be developed over numerous iterations before they are acknowledged by clients (Highsmith, 2002a).

4. Time-boxed: The main purpose of the time-box is for team members and management to set fixed delivery times for iterations of the software. This ensures that development fits into the arranged time period dedicated to that particular project, and the functionality might expand or contract accordingly (Highsmith, 2002b).

5. Risk-driven: ASD projects rely on addressing issues which have high risk through collaborating, speculating and learning to avoid risk (Koch, 2005).

6. Change-tolerant: ASD is designed to accommodate changes in each cycle, which is why Highsmith (2002) calls it a "change-oriented life cycle".

ASD is a complex adaptive structure which includes shareholders and team members, and various environments. It supports cultural elements as critical success factors such as the bureaucratic management structure, whether or not it is customer-centric, whether there is rapid communication and feedback from customers and whether there is trust between stakeholders and team members.

In conclusion ADS recognises and provides good support for organisational, people and project critical success factors which focus on project management that promotes a good relationship with the customer and is knowledgeable in agile principles and processes, and project teams where members work in a cohesive, self-organising manner, and have great motivation and are committed.

2.2.5 Lean Software Development

Poppendieck and Poppendieck (2003) define lean development software (LD) as a set of tools and principles that a software development project can use to be more lean, by changing the attitude of CEOs in an organisation. Human effort, investment and the effort to adapt to a new market environment are needed. LD is composed of seven lean principles that include twenty-two LD tools which support changing CEO attitudes (Poppendieck & Poppendieck, 2003) as discussed below:

1. Eliminate waste: This principle consists of two techniques (1) seeing waste (which focuses on looking for waste in terms of partly completed task, defects, extra features and extra processes), and (2) value stream mapping (which focuses on how much time is fruitlessly spent waiting for value to be added). (Poppendieck & Poppendieck, 2003) Figure 2.8 shows an instance of an LD project with a negligible delay time.

2. Amplify learning: Lean principles include four techniques, namely: (1) iterations which rely on iterative development to provide negotiated adjustments, (2) feedback which focuses on respondents who need to get feedback as frequently as possible, (3) set-based development, such as brainstorming the solution space and the solution that will emerge, and (4) synchronisation which is based on a mechanism to synchronise software development work. (Poppendieck & Poppendieck, 2003),

3. Decide as late as possible: This principle includes three techniques, namely: (1) making decisions (i.e. decisions must be completed focusing on lean principles), (2) at the last responsible moment (which depends on the instant at which failing to make a decision negates a significant substitute), and (3) options thinking (which relies on holding choices open pending more information becoming available). (Poppendieck & Poppendieck, 2003),

4. Deliver as fast as possible: This includes three techniques, namely: (1) cost of delay (which relies on financial models to reflect delivery date trade-offs and cost for stakeholders to be considered), (2) queuing theory (which is used to examine and remove blockages in the project), and (3) pull systems (which focus on publicly posted charts, daily stand-up meetings, and near-term targets). (Poppendieck & Poppendieck, 2003),

5. Empower the team: There are four techniques for achieving this, namely: (1) self-determination (which focuses on people and team members who are best able to improve their own process and eliminate waste in the project), (2) motivation (which relies on inspiring with purpose, safety, progress, belonging and competence, and leadership with technical management skills and project management skills), and (3) expertise (which implies being able to recognise what communal knowledge to use). (Poppendieck & Poppendieck, 2003).

6. Build in integrity: This includes four techniques, namely: (1) perceived integrity (which focuses the user's point of view), (2) conceptual integrity (which focuses on inclusion

of consistency, architecture and elegance), (3) testing (which relies on work to be done during the development process, not just at the end, to ensure system truthfulness), and (4) refactoring (which depends on restructuring a module if needed to improve integrity).

7. See the whole: There are two lean techniques to support this, namely: (1) measurements (which rely on measuring at a higher-level aggregation of data), and (2) contracts (which focus on guaranteeing successful project satisfaction and execution of predetermined responsibilities between suppliers and consumers).

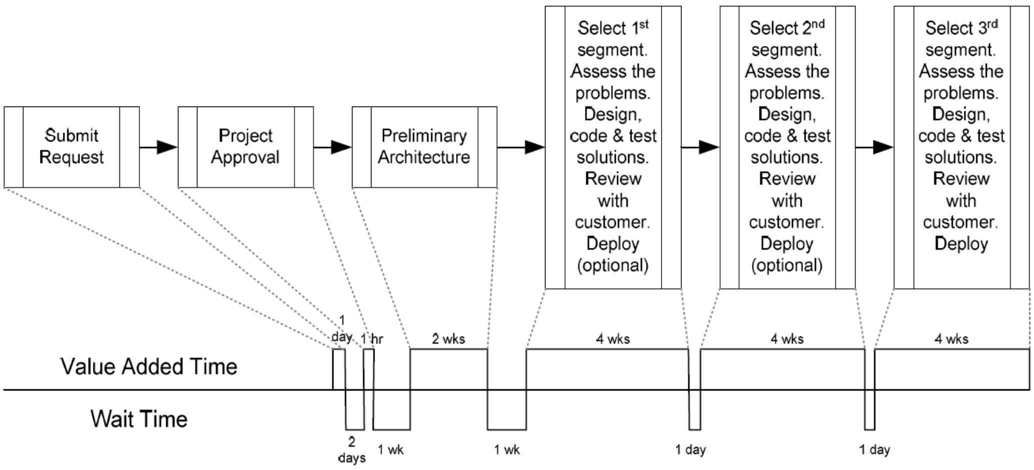


Figure 2.9: Agile value stream map from LD perspective (Poppendieck & Poppendieck, 2003)

In conclusion lean software development supports organisational, people and project factors as critical success factors.

2.2.6 Feature Driven Development

Both the definition of Feature Driven Development (FDD) provided by Larman (2004) and that provided by Misra, Kumar, Kumar, Fantazy and Akhter (2012) are used in this study. According to Larman (2004), FDD is a software development methodology which has adequate procedures to guarantee scalability and repeatability, while encouraging originality and invention. Misra et al. (2012) define FDD as a client-centric, architecture-centric and pragmatic software process. In addition, according to Koch (2005), FDD is unusual among agile methodologies due to the fact that it focuses on upfront planning and design, though it is different from traditional methodologies. FDD is regarded as agile in its iterations of the incremental structure of recognised features, and in the way the modifications in the characteristics plan and list are housed (see Figure 2.9) (Koch, 2005; Larman, 2004).

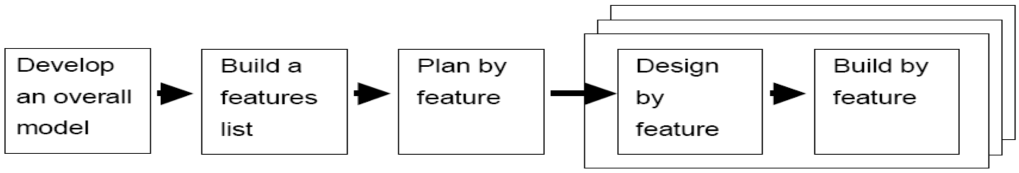


Figure 2.10: Sequential process for FDD (Abrahamsson et al., 2002)

The main aim of an FDD project is to construct a comprehensive object model of the software, with the purpose of obtaining, to the extent possible, the requirements and assumptions of the software project from the project owner (Larman, 2004). The domain object model is not static and includes corrections and new requirements that emerge during the incremental process (Misra et al., 2012).

The eight principles of FDD according to Highsmith (2002a), and Felsing and Palmer (2002) are as follows:

1. Developing by feature: Development in FDD is completed feature by feature. Each feature is small, simple and may be developed in the most optimal time (no more than 2 weeks) through a well-defined process (Misra et al., 2012).

2. Class ownership: FDD requires a project owner who possesses every object class and that they then educate other team members (Misra et al., 2012). Therefore, most features using the object classes require involvement and dedication of the owner.

3. Feature teams: In FDD, teams are formulated to develop a feature then separated when the feature is accomplished and confirmed (Larman, 2004). Every team is managed by the feature owner and includes the owner of all the object classes involved in the feature. In addition, an FDD project involves multiple feature teams working in parallel, creating and terminating software projects on a weekly or daily basis.

4. Inspections: FDD team members use rigorous inspection recommendations to discover faults, enforce coding standards and confirm knowledge of the classes by team members (Misra et al., 2012).

5. Regular build schedule: FDD includes regular builds to increase features of the software, normally weekly or daily (Misra et al, 2012).

6. Configuration management: FDD illustrates the significance of configuration management within the environment of consistent builds with various features being added which calls for careful execution of this practice. (Misra et al., 2012).

7. Reporting or visibility of results: FDD uses a reporting and tracking mechanism that mathematically calculates the project progress based on weighted features and list indicators which include design, build and code (Felsing and Palmer, 2002). Using values ranging between 0 (meaning features aren't working on) and 1 (meaning accomplished features) the feature project and value status can be calculated using the formula shown below:

$$\text{Feature value} = \text{Sum (weights of finished indicators)}$$

$$\text{Project status} = \frac{\text{Sum (Feature value) for entirety features}}{\text{Total number of features in the product}}$$

In conclusion, FDD supports organisational, people and project factors as critical success factors.

2.2.7 Crystal

The Crystal method was designed by Cockburn in 2001 and revised in 2002 and 2006 by Farhan, Tauseef and Fahiem (2009). Crystal's main distinction from other agile methodologies is that it holds that there might be a need for different conventions and policies for every project (Farhan et al., 2009). Highsmith (2002a) pointed out that Crystal enables one to select the strategies appropriate for a specific field, for example what functions for a military project might not function for a web content project.

According to Farhan et al. (2009), Crystal Clear (a variation of Crystal) allows better unity within the team and creates a more comfortable environment to work in. The Crystal methodology focuses on communication and is people-centred.

Cockburn (2002) explained the Crystal process as a process which relies on talent, interaction, people, skills, communication and community as first priority with regards to performance. According to Highsmith (2002b), the heaviness of the methodology outweighs the risks of an unreliable project. Cockburn (2002) describes the Crystal process as a set of methodologies from which team members can choose an initial point and then further modify it to the requirements of the project. According to Cockburn (2002), the methodology reflects the numerous features of a gemstone, each with a diverse face reflecting the fundamental principles and values of the methodology. According to Strode (2005), Crystal methods are refined to suit each particular project. One of the main features of Crystal is self-adaptation and incremental delivery, not exceeding more than four months. In addition, according to Farhan et al. (2009), heavy intercommunication among shareholders and light weight documentation are able to address any potential drawbacks of Crystal methodologies in almost every project.

Computerised regression testing is a distinctive feature of Crystal methods (Strode, 2005). Additionally, customers are dynamically involved in the Crystal methods (Strode, 2005). Crucial practices of Crystal include: writing test cases, pair programming, and iterative development, just to mention a few. Theunissen, Herman & Theunissen (2003) suggest that the number of team members in project or the project size determine the Crystal methodology to be chosen. The Crystal family includes Crystal Orange, Crystal Yellow and Crystal Orange Web. From the initiation of the agile alliance, Cockburn (2002) has explained how these methodologies fit into agile and how some of the other agile methodologies fit into the matrix.

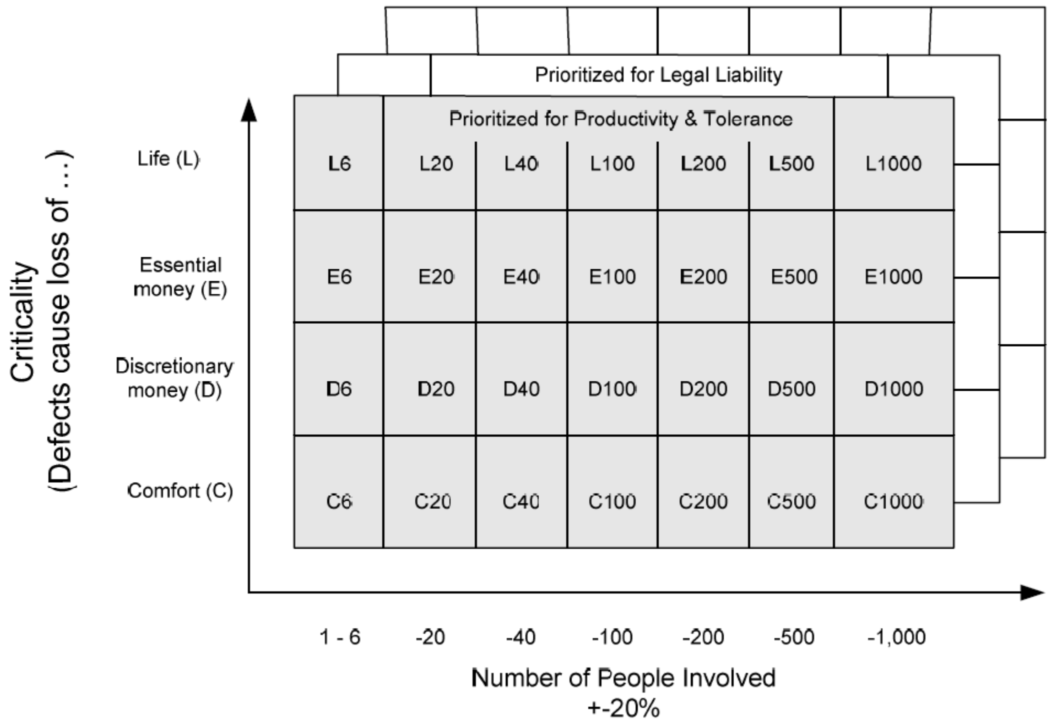


Figure 2.11: The family of Crystal methods (Cockburn & Highsmith, 2001; Highsmith, 2002b)

The factors that affect which Crystal method to select include system criticality, project priorities and communications load (which is dependent on staff size). In Figure 2.11 below, the y-axis signifies system criticality while the x-axis designates staff size, and the z-axis shows diverse project priorities such as, cost reduction, legal liability or time to market. Moving up on the y-axis signifies more critical damage that could be caused by defects in the software, and so moving up the y-axis means that more laborious and official processes are needed, while moving to the right on the x-axis means the project will need to coordinate more individuals, leading to a heavier methodology (Theunissen et al., 2003). For instance, in Figure 2.11, box C1000 will need extreme precision in the methodology, while box C6 (which refers to a small team with a non-critical project) can employ a more casual software development process.

In Crystal, the adaptation and selection of a particular methodology is the key and this is left up to the project team members (Highsmith, 2002b). The Crystal techniques execute two rules which are: 1) that the project must use incremental development cycles not exceeding four months (as mentioned earlier), and 2) that the team members need to hold reflection workshops so that the methodology can be self-adapting (Highsmith, 2002a; Highsmith, 2002b). Similar with other agile techniques, Crystal techniques are governed by self-adaptation and increments.

Therefore, Crystal supports mostly organisational factors as critical success factors with a focus on the project team working in a facility with a proper agile-style work environment, such as a dedicated office with pair programming workstations, ample wall space for postings, a communal area, no separate offices or cubicles, team members working in the same location for ease of communication, constant contact, a cooperative culture instead of a hierarchal one, a face-to-face communication style, and good management or executive support.

Each of the seven agile methodologies covered in here follow set principles which are governed by the Agile Manifesto. In addition, agile methodologies are not project specific implying that the methods can be applied to several projects.

2.3 Agile methodologies versus traditional methodologies

Agile methodologies were initiated to overcome the challenges in traditional development methodologies (Nerur et al., 2005). A group of software development professionals come up with the agile principles published as the “Agile Manifesto” as a way to overcome challenges experienced in traditional methodologies (Nerur et al., 2005). The agile principles include satisfaction of the customer needs, collaboration between team members, motivation among the team members, face-to-face conversation among the team members, adaptation to rapid change in customer requirements, and self-organising teams (Chow & Cao, 2008). As discussed above, instances of agile methodologies that support the Agile Manifesto include scrum, lean development, extreme programming, and crystal methods.

Chow and Cao (2008) explained the differences between traditional methodologies and agile methodologies focussing on two pillars of assumptions with regards to the end user. Firstly, traditional methodologies assume that end users do not know their user specifications but programmers do, while agile methodologies assume that both end users and programmers do not have a full understanding of system specifications at the initial stage (Koch, 2005). Hence, in traditional methodologies programmers need a comprehensive set of requirements in order to absolve themselves of obligation by stating that they just build the system in the way specified by the end users. In agile methodologies however, both end users and programmers learn about the system specifications as the development process evolves (Koch, 2005; Nerur et al., 2005). The second assumption made by traditional methodologies is that end users do not have enough knowledge, and thus programmers have to build in extra functionalities to meet the future needs of end users, often leading to overdesigned or failure of the software projects (Chow & Cao, 2008). Furthermore, there exist several differences in philosophy between traditional methodologies and agile methodologies which lead to differences in a number of practices and requirements, namely planning, control of the tasks, the role of programmers, the stakeholder's role, and technology used, just to mention a few.

Nerur et al. (2005), and Koch (2005) have explained some of the challenges of migrating from traditional methodologies to agile methodologies. These include management, people, process, and technology. The difficulties in migrating to agile methodologies make it doubtful that previous research into the acceptance and success of traditional development methodologies would be readily appropriate for agile methodologies (Nerur et al., 2005). For instance, factors previously found to be significant for traditional development methodologies may be insufficient in capturing the distinct features and usage contexts of agile development methodologies.

2.4 Failure factors of agile software projects

Most common failures in software projects are recognised based on the experiences of individuals (Reel, 1999). These experiences can then be generalised to similar software research projects to help others avoid similar mistakes (Cohn & Ford, 2003). According to Reel (1999) and Cohn and Ford (2003) there are 10 signs of software development project failure when focusing on generic software development projects of which at least seven are recognised before the design of the software project is developed. Larman (2004) concurs and further identifies errors and mistakes which occur in agile software development projects. Boehm and Turner (2005) further support the arguments by explaining some of the management issues related to fulfilling agile software development projects. Furthermore, Nerur et al. (2005) have outlined problems related to individuals, process and expertise in agile software development projects.

2.4.1 Organisational factors

Organisational factors include issues related to executive or top management, organisational culture, organisational size and logistics.

1. Lack of executive sponsorship or top management commitment: According to Reel (1999), lack of executive sponsorship in the transition from traditional to agile software development processes, unsuitable management arrogance, non-flexible management style and lack of management commitment significantly jeopardise the success of an agile project (Boehm & Turner, 2005; Nerur et al., 2005).

2. Organisational culture that is too political or traditional: Nerur et al. (2005) explained that organisational culture has an important influence on the social framework of organisations, which in turn impacts on the behaviour, beliefs and actions of people. They emphasize that culture impacts problem-solving strategies, decision making processes, ground breaking practices, information filtering, social discussions, associations, and planning and control mechanisms. Boehm and Turner (2005) suggested that an organisational culture that depends on traditional methodologies of developing software and operating ICT will have problems in implementing agile projects. Cohn and Ford (2003) concur, adding politics as another factor which affects the success of agile software projects. Nerur et al. (2005) also included inappropriate performance measurement and reward systems as an addition factors, since agile methodologies encourage team members by giving rewards, which might lead to political resistance amongst teams and individual actors.

3. Organisational size: Boehm and Turner (2005) explain the importance of good communication among group members who are situated close to each other. This communication typically consists of daily meetings, trust among group members, and shared tacit knowledge. They further emphasis the weakness of dividing a large group of team members into subdivisions as this causes poor communication and coordination.

4. Lack of agile-friendly logistical preparations: Cohn and Ford (2003) explain that agile methodologies need customer representatives and developers in close association, with particular emphasis on team location because a physically

distributed development team might not function well. Furthermore, Boehm and Turner (2005) suggested that specific preparation is needed for agile-oriented logistics due to the fact that there may be a need for pair programming stations, walls for progress charts and obligations. These structures permit group members to share data in a meaningful way, enabling, for example, constant integration and regression testing of the software project.

2.4.2 People factors

People factors include lack of necessary skills, insufficient project management competence, lack of teamwork and bad customer relationships.

1. Lack of necessary skills: According to Cockburn and Highsmith (2001) and Reel (1999) failure of a software development project can be caused by insufficient technical skills. Cohn and Ford (2003), and Boehm and Turner (2005) concur, adding that rigorous techniques such as continuous testing and pair programming, can help address this situation.

2. Lack of project management competence: Agile project managers are required to play a multi-purpose role in order to be successful in their projects (Cockburn & Highsmith, 2001; Cohn & Ford, 2003). This includes playing the pivotal roles of training and leading. On the same note, insufficient management knowledge and competence will lead to failure as management must be aware of errors which occur when established agile software development project practices are overlooked (Nerur et al., 2005; Cohn, 2009; Cohn & Ford, 2003; Reel, 1999).

3. Lack of teamwork: Cooperation is pivotal in all software development projects, but especially in agile projects. Software projects without cooperation among team members will lead to failure (Nerur et al., 2005). Developers who lack team work skills will meet resistance from other group members or individuals working on the same software projects (Boehm & Turner, 2005; Cohn, 2009; Reel, 1999). Agile software projects can be affected by a single rebel in the team, which can render the agile software development project ineffective and inefficient (Larman, 2004).

4. Poor customer relationships: Nerur et al. (2005) explain the advantages of agile team members working closely with their client representatives throughout the project. They further suggest that good customer relationships are significant and need customer commitment, trust, respect and knowledge. Any misconduct among project team

members, management and customers will cause the project to be unsuccessful (Larman, 2004).

2.4.3 Process factors

Process factors include ill-defined project needs, scope and planning, shortage of agile progress tracking mechanism, and lack of customer presence or ill-defined customer roles.

1. Ill-defined project needs, scope and planning: Cockburn and Highsmith (2001) and Reel (1999) have noted that the accomplishment of a software development project might be difficult if the project scope is ill-defined. In addition, Boehm and Turner (2005) mentioned the significance of clear project requirements in agile projects and noted that situations which are faced when requirements are too informal can cause difficulties for software engineering validation functions. Reel (1999) concurs with Boehm and Turner (2005) saying that planning is needed for the schedule of the project to be successful. Larman (2004) suggested the need for predictive planning instead of adaptive planning in agile projects which will have challenges along the way to project completion.

2. Shortage of agile progress tracking mechanisms: According to Cohn and Ford (2003), executives and managers find it difficult to monitor progress on agile progress as compared to traditional projects where the manager asks, for example, whether the necessary documents have been produced (Cohn & Ford, 2003). It is difficult to apply techniques used in traditional projects to agile projects, such as the rapid-pace progress measurement techniques. Boehm and Turner (2005) agree saying that traditional project progress measures are difficult to use in agile projects due to the fact that agile is governed by a task breakdown structure and flexible time boxing needs.

3. Absence of customer presence or ill-defined customer role: Agile projects need a minimum of one customer representative to be permanently located where the project is being carried out in order for the project to be successful (Larman, 2004). In cases whereby there is a large project being developed, a group of customer representatives are required to be permanently on the spot where the software is being developed (Larman, 2004). This avoids customer representatives being overworked on the developer site. If the customer's role is ill defined, for instance if the customer has no decision making power, this will cause the project to be unsuccessful (Cockburn & Highsmith, 2001; Larman, 2004).

2.4.4 Technical factors

Technical factors include lack of a complete set of correct agile practices, and inappropriateness of technology and tools.

1. Lack of a complete set of correct agile practices: Larman (2004) suggested a number of issues relating to technical traits such as incorrect pair programming, no upfront unit test design and poor integration of the quality assurance team.

2. Inappropriateness of technology and tools: Unsuitable technology will lead to unsuccessful software project. For instance, organisations that depend entirely on core technologies can find it problematic to integrate oriented development techniques. Therefore, companies need to plan to use agile methodologies by investing in tools that fund and aid rapid iteration development, configuration management and other agile techniques (Nerur et al., 2005). Cockburn and Highsmith (2001) and Reel (1999) also explain how software development projects needs to avoid challenges initiated by technology changes.

2.5 Success factors of agile software projects

Project managers tend to hold different views of success with regards to software development projects (Misra, et al., 2009). These are reflected by various authors. For example, according to Bullen and Rockhart (1981) success factors refers to a specific number of areas in which satisfactory results will ensure successful competitive performance for the individual, organisation or department, thereby permitting a few key areas where "things must go correct" for the business to flourish and for the managers’ vision to be attained. Therefore, the scope of the project and human behaviours are some of the key attributes of the success of a project (Bullen & Rockhart, 1981). On the other hand, Schwaber and Beedle (2002) pointed out that time, cost and quality are the key components of any project success (Schwaber & Beedle, 2002). Software development project managers use time, cost and quality as main competitive criteria in their day to day activities. The main objective of software development is therefore to develop high quality software with less time and cost (Misra et al., 2009).

Various studies with regards to the success factors of agile development software (such as Misra et al. (2009), Bullen and Rockhart (1981), Chow and Cao (2008), Dyba and Dingsøyr (2009), Subiyakto and Ahlan (2013) and Reel (1999)) have suggested the following success factors for agile software development projects: organisational factors, people factors, process factors, technical factors, human social factors, and project factors. According to this researcher, the following additional factors should be included: effort expectancy, performance expectancy, social influence, actual success and intended behaviour towards the success of the agile software development project based on the above literature review.

Due to the different views on what constitutes success and the incompleteness of the existing list of success factors there is a need to develop the main themes and concepts that underlie success factors for agile software projects. This research is based on the critical success factors and research models of Chow and Cao (2008), Misra et al. (2006), Misra et al. (2009), Chan and Thong (2009), Lee and Xia (2010), Wan and Wang (2010), Jun, Qiuzhen and Qingguo (2011), Sudhakar (2012), Sheffield and Lemétayer (2013), and Joseph (2013).

Themes of success factors of agile methodologies

Figure 2.12 below shows the success factors of agile methodologies broken up into themes, namely organisational factors, technical factors, social factors, political factors, people factors, process factors and cultural factors and the relationships between them. These are the central concepts in the design of the theoretical framework.

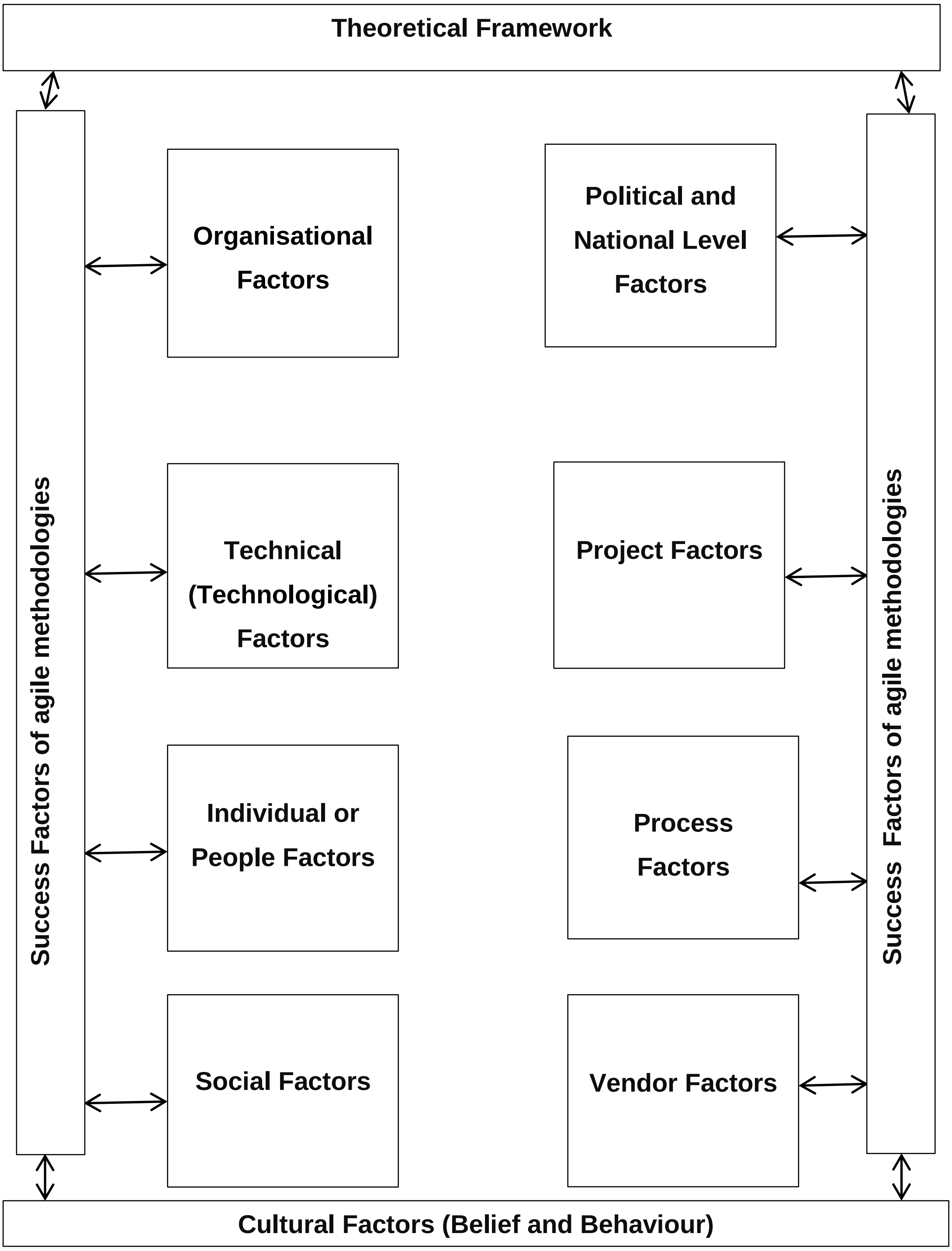


Figure 2.12 Theoretical framework of success factors of agile development software projects (Source: Own)

The following four criteria used to determine the success of agile software development projects are based on Chow and Cao (2008), Lee and Xia (2010), Misra et al. (2006), Misra et al. (2009), and Chan and Thong (2009).

- Quality (delivering a good product or project outcome),
- Scope (meeting all requirements and objectives),
- Time (delivering on time), and
- Cost (delivering within the estimated cost and effort).

Better quality software projects can be attained if there is more budget and time at disposal (Misra et al., 2006). However, project managers are usually constrained with respect to both time, budget, and the quality of the software the project is expected to deliver (Misra et al., 2009). The same expectations are still effective for software projects using agile methodologies (Chow & Cao, 2008). Hence, any software project using agile practices would be considered to have succeeded if it can deliver better quality software in a shorter time, and lower budget than traditional software development practices (Chow & Cao, 2008).

Social Influence

Cabrera, Collins and Salgado, (2006) found that social influence had a significant impact on the success of agile software development projects. If programmers believe that they have the ability to use an agile methodology, they are more likely to use it for software development, and engage in associated knowledge management activities which may not be specified by the job requirements. Hence, programmers with high social influence will be able to take full benefit of the agile methodology to perform their tasks, and also find it easier to use as shown in figure 2.12.

Culture Factors

Culture is formed as users become familiar with their surroundings. Familiarising themselves with their settings, they are able to address common social challenges and adopt the essentials that lead to success (Boehm & Turner, 2005; Triandis, 1980a). Consequently, the organisational culture and the particular social community in any organisation vitally impact on users' attitudes and behaviours towards learning and regulating new inventions in the organisation (Koch, 2005; Triandis, 1980). Therefore, users within various cultural settings will have diverse behavioural intentions, meaning attitudes towards the success of agile as shown in figure 2.12.

Political Factors

Politics are external factors that have previously been argued to have a positive direct impact on the success of the agile software development projects (Koch, 2005; Venkatesh et al., 2003). For example, if agile professionals continuously support junior software developers and train them, the organisation will in turn assist other junior software developers to attain self-efficacy with the scheme. Furthermore, it is presumed that political impacts are entrenched in the users' day to day events. That means that social events are not only affected by the people they perceive to be significant in their specific job activities, but moreover by their environments (Venkatesh et al., 2003).

Organisation Factors

Numerous researchers, such as DeLone and McLean (2002) and Curtis and Payne (2008), suggest that organisational factors play an important role in the use of agile methodologies which ultimately leads to the success of agile software development projects. Similarly, research on agile methodologies showed that some of the top critical success factors of agile methodologies are organisational factors (Koch, 2005; DeLone & McLean, 2002). Therefore, scientists have found that top executive or management support is one of the first priorities with regards to agile methodologies (Chow & Cao, 2008).

People Factors

According to Triandis (1980b), people's beliefs play a significant part in whether users believe a system is easy or difficult to use. This explains why, when people believe that a system is easy to implement, they will implement it, and if not why they will avoid it (DeLone & McLean, 2002). So when users perceive a system as easy to use and they go ahead and use it. This means that it is also easier for them to understand its benefits. Likewise, users' viewpoints have an effect on their societal beliefs. Bossini and Fernández (2013), and Boehm and Turner (2003c), have noted that, individual or people factors are critical success factors affecting software projects using agile methodologies.

Technology Factors

This research shows that technological factors can have direct influence on the success of the agile software development project respectively (Marnewick & Labuschagne, 2009; Augustine, Misra et al., 2009; Ceschi, Sillitti, Succi & De Panfilis, 2005, Venkatesh et al., 2003).

Process Factors

This research shows that process factors can have direct influence on the success of the agile software development project respectively (Marnewick & Labuschagne, 2009; Augustine, Misra et al., 2009; Ceschi, Sillitti, Succi & De Panfilis, 2005, Venkatesh et al., 2003).

Project Factors

This research shows that, project factors can have direct influence on the success of the agile software development project respectively (Marnewick & Labuschagne, 2009; Augustine, Misra et al., 2009; Ceschi, Sillitti, Succi & De Panfilis, 2005, Venkatesh et al., 2003).

Vendor Factors

This research shows that, vendor factors can have direct influence on the success of the agile software development project respectively (Marnewick & Labuschagne, 2009; Augustine, Misra et al., 2009; Ceschi, Sillitti, Succi & De Panfilis, 2005, Venkatesh et al., 2003).

2.5.1 Organisational factors

Organisational goals are achieved by implementing the organisational strategies that are derived from its mission, goals-guiding principles, and objectives. Organisational factors include the actions and decisions made regarding location and exploitation of resources to achieve organisational goals and objectives (Al Tamimi, 2014; Almahamid, 2013; Goh, Pan & Zuo, 2013; Huisman, Nd). These actions and decisions affect software development and have been identified as one of the main success factors of agile methodologies. (Al Tamimi, 2014; Goh et al., 2013).

In this study, organisational factors refer to those factors related to an organisation that may affect the success of agile methodologies. There are many organisational factors that can influence agile methodologies (Chow & Cao, 2008; Huisman, Nd). These originate from many sources within an organisation including administrative policies and strategies, the structure and design of the organisation, and organisational processes and working conditions. Other empirical findings show that organisational factors, such as top management support, effective project management and training, and change of management have a significant influence with regards to the success of an agile software development project (Chow & Cao, 2008; Huisman, Nd). From the reviewed literature, the following factors were identified as critical organisational factors: top management support, clear organisational goals, objectives and scope, a project champion, business process re-engineering, change management, interdepartmental cooperation and communication, and project management (Almahamid, 2013; Chow & Cao, 2008; Goh et al., 2013; Huisman, Nd). The study will focus on: sponsor commitment, executive or top management, organisational culture, logistical issues and suitable reward system, and customer centric matters.

1. Sponsor commitment, executive or top management: Agile software development projects are new in most ICT organisations in South Africa, which makes these project more vulnerable to failure due to lack of experienced staff, including management (Joseph, 2013). Executive or top management support and training are required for the success of the project, as these are key for proper planning and controlling the use of agile methodologies (Choi, 2000; Schatz & Abdelshafi, 2005; Yew Wong, 2005; Yew Wong & Aspinwall, 2005). Therefore, changes in top management are not encouraged. In addition, a successful project requires dedicated people, sponsors who are involved directly with the software project and people who are enthusiastic to put everything into practice, confront resistant, and encourage management and employees towards the organisation's mission and vision through communication (Cohn & Ford, 2003; Moffett, McAdam, & Parkinson, 2003; Schatz & Abdelshafi, 2005).

2. Organisational culture: Culture defines the core beliefs, values, norms and social customs of an organisation (Rogers, 1983). Agile projects will succeed in an organisational culture that is agile-friendly. This implies that agile software development projects will succeed in a culture where individuals feel contented and empowered by having liberty (Boehm & Turner, 2003c). In addition, the agile project will succeed in a cooperative culture, as opposed to a hierarchal culture (Koch, 2005; Reifer et al., 2003). Agile projects are also more likely to succeed when the agile methodologies are universally recognised within the organisation and perceived by software development experts to be a necessary factor in achieving success (Karlstrom & Runeson, 2005; Lindvall et al., 2002). A dynamic and fast changing organisation will find agile software development methodologies appropriate (Abrahamsson et al., 2002). Since agile practices rely on customer feedback and collaboration, the organisational culture needs to be supportive by nurturing a collaborative environment in the organisation (Boehm & Turner, 2003d).

3. Proper agile logistic preparation: Proper agile logistic preparation comprises two traits which are the logical preparation and the physical preparation.

    - The logical preparation requires that the organisation's management ensure that everybody on a team remains co-located, enabling face-to-face communication (Ambler, 2009; Cohen, Lindvall & Costa, 2004; Reifer et al., 2003). Group size also has an influence on communication between group members. A group that has a large number of members has difficulties in maintaining rapid and informal interaction (Ambler, 2006). Informal communication challenges can be resolved by use of small groups rather than large group which hinders fast communication and decision making (Cohen et al., 2004). Lastly, interaction as the group increases is an important concept to be considered as the organisation grows.

- The physical preparation requires that the facilities needed to accommodate agile-style work are in place, such as a communal area, pair programming stations, and no dispersed workspaces. (Koch, 2005). Businesses involved in dispersed international projects will be influenced by the political and cultural circumstances in particular regions or areas where they operate (Chow & Cao, 2008). The geographic dispersal and the location of the group members are significant because the local cultural, political, behavioural habits and circumstances significantly influence the efficiency of the project group (Chow & Cao, 2008). If the project involves dispersed groups, the organisation needs to have the appropriate teleconference equipment to facilitate daily meetings, thereby improving communication (Koch, 2005).

4. Suitable reward systems: Any organisation has a reward system in a specific form or shape which will acknowledge performance that appears to promote to the organisation's achievements (Koch, 2005). A fruitful agile software development project will have a reward system which is different from traditional approaches and which supports agile performance (Koch, 2005). Such a reward system can distinguish both individual and group contributions (Boehm & Turner, 2005).

5. Customer-centric matters: The agile manifesto supports customer collaboration as a significant requirement for successful agile software development projects (Boehm & Turner, 2005). Agile principles place great emphasis on attaining customer satisfaction through continuous and early delivery of valuable software (Fowler & Highsmith, 2001). Therefore, the customer needs to be on-site, active, highly motivated and have responsibilities in the project. These are generally referred to as customer commitment.

2.5.2 Technical factors

Technical factors influence or show skill in a specialised knowledge area relating to agile techniques or proficiency in practical agile skills; or requiring special awareness to be applied when using agile methodologies with regards to software projects (Iivari & Iivari, 2011; Rogers, 1983; Sutharshan, 2011; Sutharshan, 2013). In the context of agile methodologies, technical factors relate to the operation of software projects, or techniques that may be applied when developing software projects that will lead to successful technical performance (Rogers, 1983; Sutharshan, 2011). Several studies, such as Chow and Cao (2008), Parsons et al. (2007) and Huisman (Nd), have indicated that the technical aspects of agile methodologies highly influence their successful implementation. The success factors and their comparative significance might differ depending on the environment, characteristics, size and structures of an organisation (Chow & Cao, 2008). Various researchers, such as Parsons, Ryu and Lal (2007), Rogers (1983) and Sutharshan (2013), have agreed that the level of skill or expertise required by the individual depends on which agile methodology an individual is using. They also assert that the software project depends on the programmers' perception of which agile methodologies to use (Rogers, 1983; Sutharshan, 2013). In addition, the programmers' perception of the software as easy to develop or complex affects success of the project (Rogers, 1983; Sutharshan, 2013). Many of these studies recommend that organisations should ensure that they give technical support to keep users abreast of developments in agile methodologies, and gain skills and expertise in order to solicit positive beliefs that will lead to the successful practice of agile methodologies.

Technical factors include proper agile software engineering practices, appropriate technical training, and correct integration and delivery strategies.

1. Proper agile software engineering practices: Agile software engineering practices lead to success if done appropriately for a particular software project in both rapid and slow change environments. This is different from traditional methodologies which favour slowly changing environments (Sutharshan, 2013). Proper practice in agile projects includes activities such as initiating at the right time with coding standards and a planning game, followed by a simple design for each individual iteration, and rigorous refactoring throughout the software project (Highsmith, 2002a). Proper agile software engineering practices also mean planning documentation work as tasks like any others, but planning just the right amount of documentation for agile practice purpose (Karlstrom & Runeson, 2005).

2. Appropriate technical training: Pilot software development projects which are moving from traditional methodologies to agile methodologies require suitable technical training for team members on individual roles, development processes and agile skills to enable the project to be successful (Fowler & Highsmith, 2001; Koch, 2005).

3. Correct integration and delivery strategies: Correct integration of the software project is crucial for the whole project to function properly (Ambler, 2006). Therefore, team members need good communication between themselves, management and customers. This will lead to a correct delivery strategy, namely delivering regularly and delivering the significant features first, as well as a correct testing strategy, namely automatic testing of dependent functionality, and an integration strategy, namely continuous integration, which all contribute to agile project success (Ambler, 2006; Karlström, Runeson & Norden, 2005).

2.5.3 People factors

The main aim of agile methodologies is to empower people by supporting realistic goals, ownership, flexibility and shorter feedback cycles (Cockburn & Highsmith, 2001). However, software developers who focus on agile methodologies often have little consideration for the non-technical aspects, such as individual factors or people factors (Boehm & Turner, 2003c; Bossini & Fernández, 2013; Cockburn & Highsmith, 2001; Lalsing, Kishnah, & Pudaruth, 2012; Sutharshan, 2013). Over time there has been an exponential growth in the number of non-technical users (Lalsing et al., 2012). This has resulted in organisations implementing agile methodologies having to consider not only the technical and functional side, but also the human factors side (Bossini & Fernández, 2013; Cockburn & Highsmith, 2001). This is because organisations employ individuals from different groups, with diverse ethnic, social and cultural backgrounds. These employees have different managment styles, experiences, and cultural beliefs that influence their performance at work (Boehm & Turner, 2003d). Several researchers, including Sutharshan (2013), Lalsing et al. (2012), Highsmith and Cockburn (2001), Bossini and Fernández (2013), and Boehm and Turner (2003c), have noted that, individual or people factors are critical success factors affecting software projects using agile methodologies. These comprise trust, conflict, commitment, cooperation, communication, dependency and satisfaction among individual group members, managers, whole groups, sponsors and customers, as discussed below.

1. Individual motivation and expertise: Agile projects need expert individual team members with problem solving abilities during programming (Ceschi et al., 2005; Koch, 2005; Sillitti, Ceschi, Russo, & Succi, 2005). According to Lindvall et al. (2002), at least 25% of the team should be experts. Past experience in developing a software project often assists when doing similar development and is considered important in an agile project in order to deliver software according to the requirements of the customers. In addition, a group member with high motivational skills is required to contribute their

expertise in the rapid changing environment for the success of the project (Ceschi et al., 2005).

2. Educated, light-touch and adaptive management: Successful agile software development projects require leaders who are knowledgeable in agile methodologies or who have a tacit interactive understanding (Boehm & Turner, 2005; Ceschi et al., 2005). Training and learning are key issues which need to be continuous addressed by team members since no one knows how much training and learning is required for successful implementation of agile projects. Therefore, agile software development project practices require less formal training, particularly in extreme programming, because in pair programming team members obtain knowledge from more experienced team members (Boehm & Turner, 2005). Mentoring and professional consulting are more useful in agile software development than formal training so that tacit understanding can be imparted among individuals. Furthermore, agile projects need a light-touch leadership style or an adaptive management approach from a knowledgeable person who knows the project's drawbacks and benefits (Augustine et al., 2005). This enables them to communicate seamlessly between the project’s various sections and steer them in the direction of continuing education and adaptation (Augustine et al., 2005).

3. Coherent and self-organising teamwork: Coherent teamwork is a key factor in software development projects, particularly in agile projects. Successful agile software development projects consist of coherent self-organising groups with good communication capability (Karlstrom & Runeson, 2005; Reifer, 2003). The team work approach considers all members as skilled and valuable stakeholders in the team (Augustine et al., 2005). In addition, self-organising teams depend on the ability of the collective independent group for problem solving mechanism and reduces forthright planning, emphasising instead adaptation to varying conditions (Augustine et al., 2005). Not only technical skill is important, but also the competency of the team members and their personal characteristics, such as collaborative attitude, honesty, readiness to learn, sense of responsibility and ability to work as a team.

4. Good customer bond: A good relationship with the customer is recognised as a significant factor in successful agile projects (Ceschi et al., 2005). Larger agile software development projects involve several customer representatives (Reifer, 2003). Therefore, a coalition agreement of on-site customers is fundamental to the success of the project, with the project group successfully creating well-known points of contact

for particular items to ensure that the project keeps moving on the right track (Reifer, 2003).

5. Agile-appropriate human resource policies: Planning for agile projects needs to include human resource concerns, for instance: promptness to time, position descriptions, individual rewards versus team-oriented rewards, and necessary skills (Lindvall et al., 2002). Agile development group participants frequently don't align with standard development position descriptions and agile job descriptions need to meaningfully include additional experience and skills to enable them to sufficiently achieve the goal (Lindvall et al., 2002). The human resources division needs to encourage people to follow non-traditional methodologies. This will include officially reconsidering policies.

6. Societal and cultural factors: Individuals must be eager to learn from each other, collaborate, be honest and responsible (Lindvall et al., 2002). For example, if a society has communicative individuals, and they are broadminded and dynamic by their nature, their work customs will be influenced as well.

2.5.4 Social factors

Social factors relate to an individual's internalisation of their societal norms (Ajzen & Fishbein, 1980; Rogers, 1983). According to Ajzen and Fishbein (1980), subjective culture, also referred to as subjective norm, involves an individual's experience, beliefs, attitudes, ideals, rules, norms and values, in relation to the individual's environment. Various studies, such as Ajzen and Fishbein (1977) and Rogers (1983), have referred to social factors as social influences. The concept of social factors includes several factors such as senior management commitment to the implementation of the critical success factors of agile methodologies (Whitworth & Biddle, 2007a; Whitworth & Biddle, 2007b). In addition, pressures from the different parties that interact with agile methodologies are involved as well (Whitworth & Biddle, 2007b). Many researchers, such as Livermore (2008), Whitworth and Biddle (2007a), Chow and Cao (2008), and Livermore (2008) have recommended that for the successful implementation of agile methodologies to occur, coordination and monitoring of the social influences of different employees (both from outside and from within the organisation) must be clearly monitored, including expert developers, organisational management and team members.

Individual social characteristics that influence usage of and intention to use agile methodologies include assertiveness, experience, practical knowledge and skill (Agarwal & Prasad, 2000). Individuals with positive assertiveness and experience will embrace the new practices easily and faster, and contribute to increased productivity (Williams, Layman &

Krebs, 2004). Users with high levels of practical skill in terms of knowledge of different application of software projects, and other related software development methodological practices, may not be subject to the learning curve related with an unskilled domain (Williams et al, 2004).

2.5.5 Political factors

Several organisations, particularly software development companies, select their top management politically (Iivari & Iivari, 2011; Sutharshan, 2013; Wan & Wang, 2010). Such political appointees have an influence over software development staffing, particularly those in executive positions. As a consequence, numerous recruited staff members are below the professional level of the standard required. However, several researchers, such as Wan and Wang (2010), Sutharshan (2013), and Iivari and Iivari (2011), urge that agile projects need staff that are experienced, knowledgeable and have the necessary skills.

It is significant to remember that external and internal political factors influence inter-functional co-operation (Sutharshan, 2011; Sutharshan, 2013). Due to the fact that agile methodologies lead to changes in business process, and requests for sharing of information, as a prerequisite for cooperation, certain managers might see this as a threat to their functional operation of the business (Wan & Wang, 2010). Hence, this might lead to an interruption of the entire integration efforts of software projects. Therefore, it is important to include political factors amongst the critical success factors of agile methodologies. These political factors influence people factors, process factors and vendor factors as themes within the theoretical framework as shown in Figure 2.1. Internal and external political factors include governmental policies, infrastructure and technological availability, as well as political influence (Cockburn & Highsmith, 2001; Wan & Wang, 2010). These political factors are particularly noticeable and have particular significance in developing world countries, like South Africa

1. Governmental policies: National government or corporate policies may influence the success of software projects (Cockburn, 2006). For example, governments may enforce strict restrictions on developing software projects, exchanging information, and transferring money between countries. Such issues force development firms to modify software projects according to their use. When an agile software project is implemented in another country rather than where it was developed, its implementation success will be unpredictable. The developing countries (in the third world sense) need to have a good relationship with developed countries to use new methodologies in their software projects with relevant experienced staff from developing and developed countries.

2. Infrastructure and technological availability: There are various challenges faced by software development projects in developing countries (Cockburn, 2006; Ecuyer & Ahmed, 2016). These include inadequate IT infrastructure, less capital, lack of governmental policies, lack of funding, lack of IT experience, and lack of IT maturity. Such obstacles, if monitored in a positive way, don’t need to result in project failure.

3. Political influence: External political influences on an organisation can play a major role, especially when it is to do with issues of decision-making (Cockburn & Highsmith, 2001). Such decisions include team member and management selection, hiring of personnel and assignment of tasks. These may have a negative effect on the business due to the inclusion of inexperienced staff, which may lead to senior and middle level managers losing power and influence, but may also keep employees on their toes as they may fear for their jobs. Uncertainty may be created which may lead to poor dissemination of information that in turn affects the success of the software project.

4. Obsolescence of computing hardware and software: Obsolescence refers to the state of hardware or software being no longer needed even though it is still functioning (Cockburn & Highsmith, 2001). Hardware or software becomes obsolete when a newer version has come to the market, hence causing the one in use to be discarded or antiquated. In most cases, organisations in developing countries use software which has been discarded or are on the verge of being replaced in developed countries. Similarly, there may be resistance against using new agile methodologies to suit their business needs. When a new version of agile is discovered, developing countries may not want to take the opportunity to use them for the success of their software projects, such as in South Africa.

2.5.6 Process factors

Process factors refer to success factors that can include management processes, work scheduling, communication focus, and customer involvement.

1. Proper management processes: Successful agile projects follow clear software development management processes (Koch, 2005). A successful agile software development project requires being in agreement with the general management process of software projects, for instance configuration management, requirement management and project management but with a substantial agile alignment (Koch, 2005). For instance, for requirement management, a focus is needed on the specification of preliminary requirements, separate from the requirements or amendments which emerge as the agile project advances (Koch, 2005).

2. Good and instant communication: Immediate and good communication among team members, management and customers is critical in agile software development projects (Reifer et al., 2003). Projects with a strong emphasis on communication and with a demanding communication schedule, such as daily meetings among group managements of large projects or daily stand-up meetings among team stakeholders instead of documentation, tend to be more successful (Ambler, 2006; Reifer et al., 2003).

3. Regular work schedule: A regular work schedule maintains a high performance workers' confidence by precluding team member's burn-out, thus enabling them to sustain the pace in the long run (Highsmith, 2002a). Software projects also need to keep team members from becoming bored. Agile software development projects require at least forty hours per week to be successful (Highsmith, 2002b; Schatz & Abdelshafi, 2005).

4. Customer participation: Any agile software development project needs at least one customer representative on the team (Koch, 2005). Good customer involvement with continuous feedback to the team assures a successful agile project (Karlstrom & Runeson, 2005; Karlström et al., 2005).

2.5.7 Project factors

Project factors include project type, project size, and project risk and cost.

1. Agile-friendly project type: Criticality of the project is one of the major factors to determine whether to use agile methodologies or not (Boehm & Turner, 2003c; Koch, 2005). Cohen et al. (2004) suggested that agile methodologies are appropriate for non-life threatening or non-life critical software projects. Moreover, agile projects are successful in well-defined applications with promising requirements in an unsteady, dynamic environment, or in complex projects which require a faster schedule with constant changes in an uncertain environment (Highsmith, 2002b; Reifer et al., 2003).

2. Appropriate project size: Agile is appropriate for small project teams with well-matched small products (Augustine et al., 2005; Boehm & Turner, 2003b; Cohen et al., 2004). Appropriate team size is less than 20 members in a team (Reifer et al., 2003). Small projects that use a single, cohesive team rather than multiple distinct self-organising teams cooperating together are most appropriate for agile (Boehm & Turner, 2003b).

3. Proper project cost evaluation and high risk analysis: Proper project cost evaluation refers to the delivery of the software within an approximated cost, time, quality and effort (Ceschi et al., 2005; Cohen et al., 2004). In addition, it is important to calculate the amount of risk to evaluate if the project is suitable for agile methods (Boehm & Turner, 2003a).

2.5.8 Vendor factors

Research must authenticate the role of external shareholders and the environment that surrounds the users of the agile methodologies (Chow & Cao, 2008; Nasehi, 2013). This is due to the fact that external shareholders, such as vendors, might affect the implementation process of agile methodologies by failing to play their part, such as providing support, and applying unnecessary pressure for change that may interfere with the users' perception or social norms influencing the critical success factor(s) of agile methodologies (Chow & Cao, 2008; Nasehi, 2013). It is therefore necessary to include vendor factors as one of the critical success factors of agile. Vendor factors comprise vendor support, consultant support, system changes and upgrade.

1. Vendor support: Choosing a suitable vendor is a vital task, as good vendors are expected to play key roles in agile systems implementation. Good vendors must provide support to the implementing organisation including training, technical assistance, upgrading and providing professional consultation.

2. Consultant support: Organisations may prefer to use consultants to facilitate the implementation process who are independent of the vendor and are not part of the organisation's employee force.

3. Systems Changes and Upgrade: In order to manage agile software systems properly, the system has to be maintained at the vendor-supported levels. Agile software systems' vendors may decide to upgrade the software as may be deemed necessary. Sometimes upgrades may occur multiple times in a year or once every several years.

2.5.9 Cultural factors

Culture denotes the complex system of meanings, symbols, and assumptions about what is legitimate or illegitimate, good or bad, that inspires the prevailing practices and norms in a society (Huisman, Nd; Iivari & Iivari, 2011; Rogers, 1983; Strode, Huff & Tretiakov, 2009). This consists of a set of beliefs, customs, moral values, language and regulations shared by a group of individuals within a group (such as an organisation) (Rogers, 1983). Organisations, particularly large ones, have a variety of diverse cultures, customs, and religious beliefs, since they include people from different backgrounds (Rogers, 1983; Strode et al., 2009). Consequently, this study considers it important to include cultural factors as critical to the success of agile software development projects. Cultural factors in turn influence: organisational, technical, individual, social, political, people, process and vendor factors as suggested by previous researchers such as Iivari and Iivari (2011), Huisman (Nd), and Strode et al. (2009). The most common cultural features in agile methodologies include encouragement of rapid communication, trust among individual members, management support of the decisions of the developers and organisational encouragement for changes in software requirements.

2.6 Research Gaps

Many theories have been used to predict users' usage and acceptance of agile methodologies from the use and technology adoption perspective (Venkatesh et al., 2003; Chan & Thong, 2009). One school of thought starts with the Theory of Reasoned Action (TRA), followed by the Theory of Planned Behaviour (TPB), the Technology Acceptance Model (TAM) and the Unified Theory of Acceptance and Use of Technology (UTAUT); and the process of model building continues. All of the usage, acceptance, and intention based theories and models in this set look at the cognitive processes of individuals even though the analysis then aggregates the individual results in order to try to predict how an individual, potential end user will act regarding technology adoption. Therefore, UTAUT have not been used in the agile software development project as the recent modified model.

Several research have been carried out but the is lack of modelling techniques applied to agile software development methodology (Marnewick & Labuschagne, 2009; Augustine, Sencindiver & Woodcock, 2005; Misra et al., 2009; Ceschi, Sillitti, Succi & De Panfilis, 2005).In light of the above, there is a need to model the critical factors that influence the success of software projects that use agile methodologies in order to avoid a waste of money, time and organisational resources. Unfortunately, not all the critical success factors that affect agile software projects can be addressed at once, however modelling the complexities of software development using new probabilistic techniques presents a positive way forward.

This has led to the following research questions:

Primary research question

What are the critical success factors that influence the success of software development projects using agile methodologies?

Secondary research questions

- How do agile professionals perceive the adoption of agile software development projects in South Africa?
- What is the most appropriate theoretical framework which can be adapted to model the critical success factors of agile software development projects?
- How can critical success factors be structured into a framework that can inform agile professionals and the community?

2.7 Conclusion

This chapter focused on the literature outlining concepts of traditional methodologies, agile principles and techniques used in agile. The agile methodologies outlined here assisted in formulating the questions for data collection and hypothesis. Understanding the concepts and limitations of agile methodologies helped in structuring, contributing and formulating success and failure factors relevant to agile software projects.

The next chapter gives a comprehensive description of the theoretical framework introduced here.

CHAPTER 3: THEORETICAL FRAMEWORK AND RESEARCH MODEL

This chapter discusses the theoretical framework underpinning the research for the study. The hypotheses are then formulated from the chosen research model. The chapter first presents the theoretical background and the details of the Theory of Reasoned Action (TRA), Theory of Planned Behaviour (TPB), Diffusion of Innovation (DOI), Social Cognitive Theory (SCT), Technology Acceptance Model (TAM), Motivation Model (MM), Combined TAM and TPB (C-TAM-TPB), and Unified Theory of Acceptance and Use of Technology (UTAUT), all of which form part of the theoretical framework for this research.

3.1 Theoretical background

TRA, TPB, TAM and UTAUT have been the main theories used in research to explain the phenomena of information systems (IS) success within organisations, and these can also be applied to agile software development projects. Theoretical models have informed IS research studies which have provided valuable evidence for future design and development of information systems. Several researchers believe that these theoretical models capture the significance of IS practise and process, how environment affects the successful practise of IS, and that these factors are consistent and can be applied generally across software development projects (Venkatesh et al., 2003). As each model was developed, several arguments, drawbacks and benefits were discussed in the research community. This inspired Venkatesh et al. (2003) to empirically review eight theories and came up with the UTAUT. UTAUT presents a clear view of why people practise or do not practise technology even after it has been acknowledged. The model comprised of TRA, TPB, TAM, DOI, Motivational Model (MM), Model Combining the TAM and TPB (C-TAM-TPB), Model of PC Utilisation (MPCU) and SCT (Rogers, 1983; Ajzen, 1991).

This chapter discusses the theories that led to the development of UTAUT, which is the conceptual framework which underpins this study.

Application of previously used theorises

Existing theories have mostly been used to examine the usage and acceptance of Information Technology tools. DOI theory, TRA, TPB, UTAUT, and TAM, have been widely used for examining individual's intentions to adopt IT innovations, such as the World Wide Web and Microsoft Office spreadsheets (Chow & Cao, 2008; Iivari, and Huisman, 2007; Koch, 2005).

Several studies on agile methodologies are based on the practical use of agile methodologies and their costs and benefits, with a couple of studies on the factors affecting the acceptance of agile methodologies as part of their success (Chow & Cao, 2008; Misra et al., 2006; Misra et al., 2009; Chan & Thong, 2009). However, some studies have used the existing theories to explore the use and acceptance of agile methodologies.

Chow and Cao (2008) performed an initial investigation into individual usage and acceptance of agile software development projects using constructs from TAM, TRA, TPB, DOI, and UTAUT. In their research, the perceived usefulness and perceived ease of use scales were shown, in general, to be reliable measures in the agile software development context. Misra et al (2009) adapted actual behavioural constructs from TAM, TRA, TPB, DOI, and UTAUT to determine critical success factors for agile software development projects. Further, team environment has been found to be an important factor that would affect the agile professional's perceptions of agile software development (Chow & Cao, 2008; Iivari, & Huisman, 2007; Koch, 2005).

These studies suggest a number of organisational, people, process, culture, technology and project factors that are crucial to the successful assimilation of agile methodologies. However, these studies have not taken characteristics of performance expectancy, intention, actual success, or characteristics of effort expectancy of individuals into consideration.

Sultan and Chan (2000) discussed the features of the technology such as relative advantage, perceived usefulness, which were non-significant in determining adopters and non-adopters of Object Oriented technology. The results challenged the findings in other studies, possibly because both adopters and non-adopters were fully conscious of the expected paybacks of the technology since they are experienced programmers, and they may differ in their adoption judgments due to other factors, such as features of individual programmers and organisations.

3.2 Theory of Reasoned Action

TRA has been used by the various IS researchers in order to explain users' behaviour (Davis, 1989; Davis, Bagozzi & Warshaw, 1989). This theory is based on the intention, behaviour, subjective norm and attitude of individuals. TRA assumes that a person's belief regarding a specific behaviour will affect the person's attitude. Ajzen and Fishbein (1977) found that human attitudes toward the intention to act have a greater effect than attitude toward the actual behaviour. TRA defines attitude as a person's general favourable or unfavourable feelings about a particular behaviour (Ajzen & Fishbein, 1977). Moreover, the attitude determines the comparative strength of the person's intention to carry out that behaviour (Rogers, 2008). A person is more likely to carry out that actual behaviour if that person has a higher degree of intention. In addition, Roger (2008) argues that the intention to carry out a specific behaviour is influenced by subjective norms and attitudes. Furthermore, subjective norms are affected by normative beliefs concerning that specific behaviour. Normative beliefs refer to individuals' beliefs pertaining to the extent to which other individuals who are important to them (for example family and friends) think they must or must not perform specific actions (Ajzen & Fishbein, 1977). According to Ajzen and Fishbein (1975) a subjective norm is the "perception that most people who are important to him, think he should or should not perform the behaviour in question". TRA's construct model is illustrated in Figure 3.1.

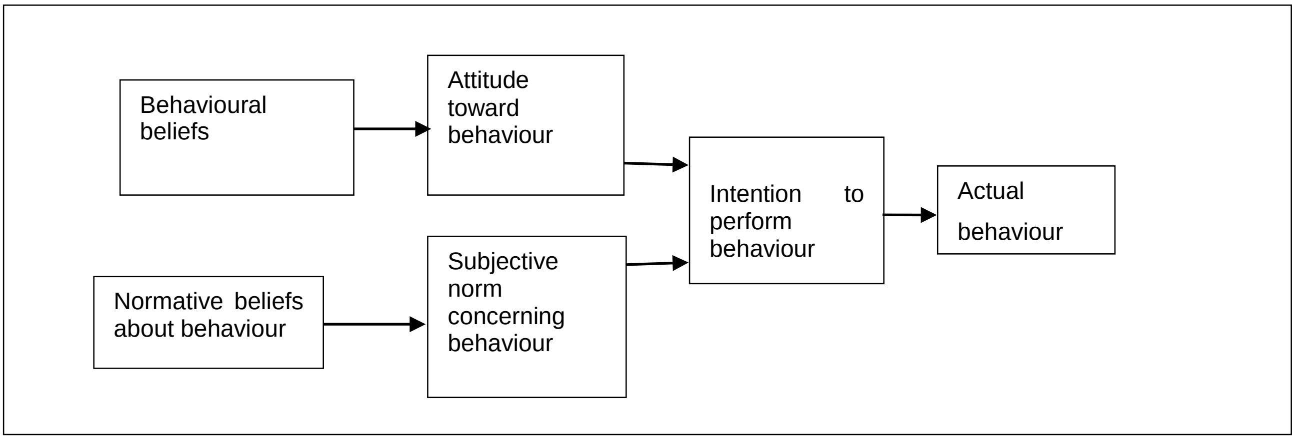


Figure 3.1: Theory of Reasoned Action (Ajzen & Fishbein, 1975)

TRA is extensively used by psychologists to predict and explain human behaviour in particular circumstances (Yeaman, 1988). A review by Hale, Householder and Greene (2002) showed that TRA has been validated by numerous researchers in the field of health, as well as in IS behaviour (Greene, Hale, & Rubin, 1997; Sparks, Shepherd & Frewer, 1995;Koch, 2005). Several researchers have commented that TRA has been used successfully in the domain of consumer behaviour to forecast users' behaviour and intention even though it has some weaknesses (Sheppard, Hartwick, & Warshaw, 1998).

According to Sheppard et al. (1998), the intention might be affected by a change in activities, time and other external factors that are not linked to the behaviour before a person performs an act. Davis (1989) and Yeaman (1988) argued that subjective norms are irrelevant as an influence on intention when using TRA in the field of IS. Furthermore, scientists have discovered that the behaviour used in TRA is restricted to the behaviour linked to individual volitional beliefs (Ajzen, 1985; Hale et al., 2002; Netemeyer, Ryn, & Ajzen, 1991; Sheppard et al., 1998). Thus, a person will normally perform a given action when they have the intention to do so. Nevertheless, there are a number of instances where the behaviour is out of the person's control or may even be unconscious. For instance, a behaviour might be involuntary or habitual or the behaviour might need skills and resources that the person does not possess. In addition, Davis et al. (1989) suggested that organisational structure and politics may only influence the person's behaviour indirectly by influencing attitude or subjective norms. As a result of these limitations TPB was formulated.

3.3 Theory of Planned Behaviour

TPB was formulated by Ajzen and Fishbein in 1980 as an extension of Ajzen's TRA. TPB added the factor of perceived behavioural control – known as the construct of self-efficiency – as a further determinant of individuals' actual behaviour and behavioural intent (Madden, Ellen & Ajzen, 1992). Following the constructs of TRA, TPB posits that a person's behaviour is determined by the intention to carry out the behaviour and that the intention is influenced by their subjective norms and attitude toward the behaviour.

Since one of the restrictions of TRA is that behaviour is not under a person's volitional control or beliefs (Madden et al., 1992). According to Ajzen (1985) some other factors, such as skills, financial status and time, also affect behaviour. Accordingly, several researchers, such as Bandura (1977) and Adams, Nelson & Todd (1992), found that a person's confidence concerning their capability to carry out a specific behaviour will directly affect their actual behaviour. To address the volitional concerns toward the behaviour, Ajzen (1985) adapted all constructs from TRA and proposed perceived behavioural control as an additional determinant of a person's intention. Control belief specifically refers to a person's belief about the availability of factors such as skills, money and time that correspond to specific behaviour (Sheppard et al., 1998).

Therefore, perceived behavioural control refers to a person's evaluation of the degree of ease or difficulty of carrying out a specific behaviour, focusing on control beliefs (Netemeyer et al., 1991).

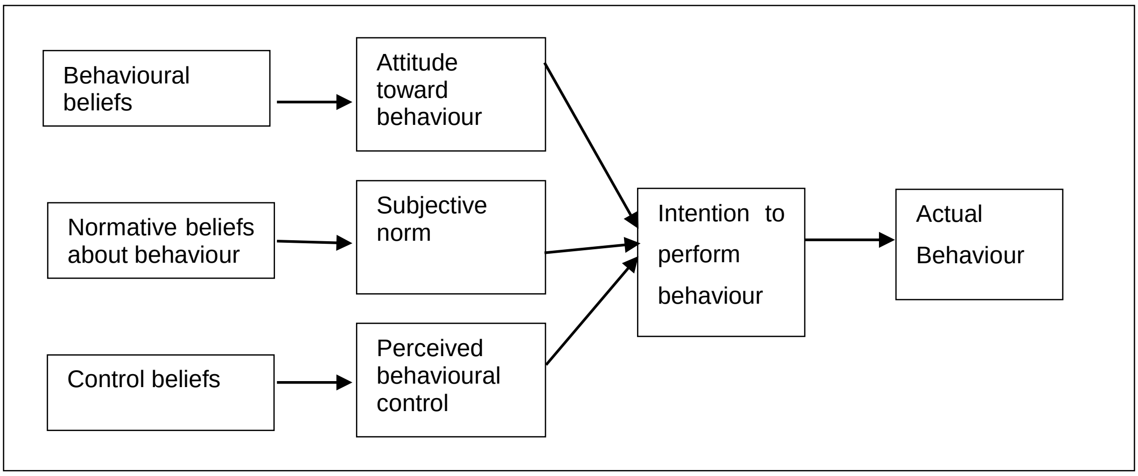


Figure 3.2: Theory of Planned Behaviour (Ajzen & Fishbein, 1980)

However, there are also counterarguments that speak against TPB. For instance, Ogden (2003 as cited by Ajzen and Fishbein (2004)) discovered that behaviour change was not reliably predictable when looking at perceived behavioural control, subjective norms and attitudes when reviewing the literature concerning TPB. Furthermore, when Ajzen and Fishbein (2004) investigated Ogden's concerns, they noted that the significance of constructs might differ across – or might even not be essentially based on – changing circumstances, behaviours and populations. In addition, according to Mathieson (1991), modified instruments are required when implementing TPB research in different contexts. According to Sharma and Kanekar (2007), TPB might be unsuitable for studies that concentrate on behavioural change since the constructs of TPB do not account for the reasons for behavioural modification over time.

3.4 Diffusion of Innovation

According to Rogers (1983), Diffusion of Innovation articulates the common model for IT adoption in the organisation structure. DOI consists of 5 phases which are: trialability, observability, complexity, relative advantage (which is regarded to as perceived usefulness in the Technology Acceptance Model), and compatibility (Adams et al., 1992; Davis, 1993; Rogers, 1983). On the one hand, Moore and Benbasat (1991) adapted the characteristics of innovations and developed a set of constructs that might be used to study customers' behaviour, namely complexity, relative advantage, visibility, image, compatibility, voluntariness of use and results demonstrability. On the other hand, Agarwal and Prasad (1998) and Rogers (1983) asserted that focussing on diffusion shifted the attention of scientists from research examining innovation changes and focussed it on individual's adoption of innovations. This approach recognises that the rate of innovation adoption is influenced by persons' perceptions of the features of the innovation rather than the actual features.

However, the DOI theory has weaknesses in that it does not take into account social, economic and political factors, especially in the organisational context (Rogers, 1962). Therefore, DOI is not suitable for complex processes, such as social interaction, in addressing the issues in IT adoption. Rogers (1983) notes that the drawback of failing to understand the human environment and organisation's context (focus) in IT adoption needs to be addressed. Therefore, Weilbach and Byrne (2010) agree with the idea of Du Plooy (1998) which suggests that for IT adoption to be successful there is a need for social and environmental perspectives to be understood by IT researchers. Accordingly, in order for IT adoption processes to be successful, they need to be based on social and technical adoption models instead of only linear technological models.

DOI provides partial confirmation on how attitude changes affect rejection or acceptance of decisions, and also how innovative features fit into this process (Rogers, 1983; Saga & Zmud, 1993). Likewise, the theory helps elucidate the role innovation features play in creating attitudes (regarded as a relative advantage in DOI).

3.5 Social Cognitive Theory

SCT was established by Bandura (1986) to provide a framework for understanding how to forecast and modify human behaviour. SCT refers to human behaviour as being linked to environment and personal factors (Goodhue, 1995). SCT emphasises that people and their behaviour are affected by their actions and thoughts (Goodhue & Thompson, 1995). Moreover, SCT affirms that the environment and the individual include cognitive competencies and human beliefs that are established and adopted by social effects and arrangements within the surroundings (Goodhue & Thompson, 1995). In addition, SCT examines the relation between the behaviour and environment, considering how a person’s behaviour defines the features of their environment and, in turn, how their behaviour is adapted by those surroundings. Compeau and Higgins (1995b) positively used SCT to forecast both individual and team behaviour, and identified methods by which behaviour can be adapted. SCT originated in imitation and social learning, and was easily extended to use and acceptance of Information Technology (Compeau & Higgins, 1995a).

3.6 Technology Acceptance Model

Davis (1989) developed TAM from TRA to forecast Information Technology acceptance and usage. TAM empirically confirmed and predicted that a person’s intention to practice technology can be affected by their perceived ease of use and the perceived usefulness of that specific technology. TAM’s conceptualisation of attitude regards it as being an outcome of two primary perceptions: perceived usefulness and perceived ease of use leading to intention of use and actual use (figure 3.3). Perceived ease of use is defined as the degree to which a potential adopter views the usage of the target scheme to be comparatively free of effort (Davis et al., 1989). Schemes that are perceived to be easier to use and less complex have a higher probability of being accepted and used by potential users (Adams et al., 1992). Perceived usefulness, referred to as the user's subjective likelihood of using a definite application system, will increase his or her work performance within an organisational setting (Adams et al., 1992; Davis et al., 1989). Both perceived ease of use and usefulness are subjective concepts and not inherent attributes of the scheme, and can be perceived inversely by diverse users.

Various researches have replicated, used, extended and validated TAM, and as the model has developed it has become evident that numerous variables impact behavioural intention, perceived ease of use, and perceived usefulness (Goodhue, 2007; Moore & Benbasat, 1996). The most popular external factors include compatibility, system quality, computer anxiety, computing support, experience and enjoyment (Goodhue, 2007).

Various drawbacks have been noticed by numerous researchers considering TAM in the context of the field of ICT (Bagozzi, 1992; Davis, 1989; Davis et al., 1989). According to Lucas and Spitler (1999), several studies that have used TAM lacked a clear difference between the objective measure of use and the measure of perceived use. Different researchers have pointed out that customers can do tasks in distinctive ways and that the technology can permit diverse ways of performing a specific duty. Lastly, TAM's explanatory capability is restricted and most of its study has been investigated in workrooms using single-function expertise rather than a variety of technologies which relate to the real life circumstances of an organisation.

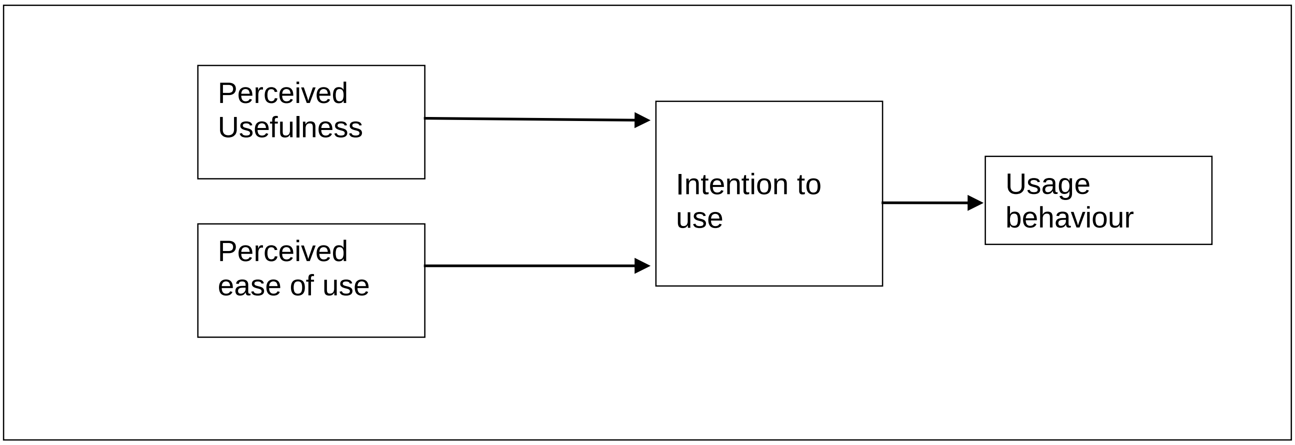


Figure 3.3: Technology Acceptance Model (Davis, 1989)

Benbasat and Barki (2007) hold that TAM has been a fruitful theory in the arena of IS study, and it has made significant impact. Benbasat and Barki (2007) point out that the effort by numerous researchers to come up with diverse versions of the model has not only initiated misunderstanding, but also severe theoretical disorder in the IS arena of research as no one form has been commonly acknowledged. This has led to the formation of MM and Combined TAM and TPB

3.7 Motivation model

Motivation is the strength that pushes a person to do a specified act or behaviour (Netemeyer et al., 1991). It is usually enforced by feelings such as drive, initiative, persistence, and intensity that inhibits, promotes, or neutralises goal-directed behaviours. The Motivational Model (MM) demonstrates that general motivation theory, and extrinsic and intrinsic motivation, is an explanation for behaviour (Madden et al., 1992). Extrinsic motivation is the perception that the individual will want to act due to the fact that they perceive the action to be instrumental in attaining valued results which are distinct from the activity itself (Davis, Bagozzi & Warshaw, 1992). Intrinsic motivation is the perception that the individual will want to act for no apparent reason other than the process of performing the act itself (Davis et al., 1992; Netemeyer et al., 1991). Davis et al. (1992) developed the Motivation Model in the IS domain. Their findings showed that persons who perceive that they are unable to perform a task will select not to act, even though they could have achieved valued outcomes if they had acted. In their application of the model to technology acceptance and use, they discovered that people who perceive technology as problematic to use will be inclined to shy away from it. According to Davis et al. (1992), extrinsic motivation occurs where a person needs to perform actions because they perceive that they will gain benefits that are outside the action itself, such as pay, promotions or performance expectance. In contrast, they define intrinsic motivation as the power that drives users to do an activity despite the fact they have no perception of future, indirect paybacks (Davis et al., 1992).

3.8 Combined TAM and TPB

According to Taylor and Todd (1995b) the C-TAM-TPB model associates the forecasters of TPB with the perceived usefulness factor from TAM to develop a hybrid model. The subjective norm (from TPB), attitude toward behaviour (from TPB), perceived usefulness (from TAM) and perceived behavioural control (from TPB) constructs were included. Through its crossbreed nature, the model usually achieves its goal of predicting customers' behaviour.

However, the model's boundaries lie in the fact that the research which led to its development relied on a student setting where perceived behavioural control and subjective norms can function inversely. Furthermore, throughout the study the students' demographics were not included in the final analysis; however controlling factors (moderating effects) in the student study have a high impact on behaviour intention (Venkatesh et al., 2003). It is also important to remember that the significance of the research results were below levels set by some research guides. Finally, the assessments of experience with regards to the study were measured on a dichotomous scale and offer only gross differences. That means that perceived ease-of-use in the study relied mostly on users without prior experience and perceived usefulness mostly on experienced users (Taylor & Todd, 1995b).

3.9 Model of PC Utilisation

The Model of PC Utilization was adapted from Triandis' (1980b) theory of human behaviour and grants a competing viewpoint as compared to TRA and TPB. Thompson, Higgins and Howell (1991) developed MPCU to predict PC operation. The following factors served as the rudimentary decision variables with regards to MPCU: complexity, job-fit, impact towards use, long-term consequences, facilitating conditions and social factor (Thompson et al., 1991). Researchers left out behavioural intentions due to the fact that the research experts were interested in reality rather than predictive use. Additionally, habits were excluded due to the challenges of measuring these. The contexts in which the study was undertaken enabled the model designers to clearly distinguish how supercomputers are used at places of work, thus making job fit and complexity key decision variables. Their research participants were restricted to those workers who used supercomputers voluntarily at work, such as managers and professionals. The study's findings specified that complexity; social factors, long-term consequences, and job fit had an important impact on user PC. The study's explanation of results shows that professionals and managers use PCs as tools therefore the affect issues do not arise whereas they attributed to the poor performance of the facilitating conditions to the measurement issues.

It is vital to acknowledge that the MPCU founding study only focused on circumstances where use was voluntary and did not consider circumstances where users are mandatorily made to use supercomputers. In principle, this makes the model well-matched for estimating persons' use and acceptance of a variety of ICT.

3.10 Unified theory of acceptance and use of technology

According to Venkatesh et al. (2003), UTAUT's objective is to elucidate user intentions to use an IS and the resultant habitual behaviour. The UTAUT model combined diverse viewpoints in the arena of knowledge acceptance research. Venkatesh et al. (2003) claimed that UTAUT tried to clarify the decision variables that are applicable in studies on the intention to use technology by using three intentional beliefs: effort expectancy, social influences, and performance expectancy. In addition, UTAUT makes provision for external variables' influence on technology use behaviour through the facilitating condition decision variable. Venkatesh et al. (2003) in their study, recognised that in order to impact user behaviour, effort expectancy, social expectancy and performance expectancy must be facilitated by intention and moderated by voluntariness of use, age, experience and gender. Therefore, the facilitating condition variable, when moderated by experience and age, directly affects use behaviour. UTAUT was able to predict acceptance and use based on the dependent variables 70% of the time compared to the 40% of TAM (Cody-Allen & Kishore, 2006). Figure 3.4 shows the interrelation of UTAUT's decision variable and its controlling factors.

UTAUT thus offers an improved theoretical framework for understanding the critical factors associated with effective usage of agile methodologies in South Africa companies. Its controlling factors fit in well with regards to team members, customer demographics and being customer-centric. Using the UTAUT model, a decent theoretical framework for the success factors of agile methodologies can be established. Although UTAUT is a single-level model, it is appropriate for examining the research hypothesis of this study. Individual constructs of UTAUT are discussed, as they relate to agile, in more detail below.

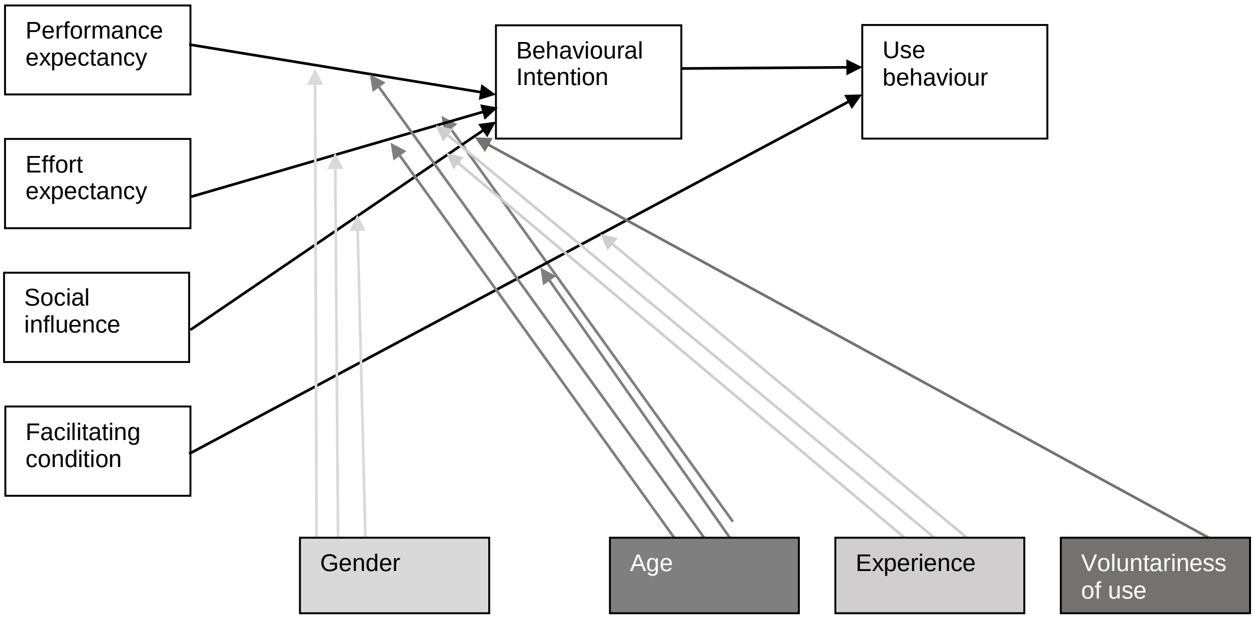


Figure 3.4: The UTAUT model (Venkatesh et al., 2003)

3.10.1 Performance expectancy

Relating UTAUT's constructs to agile methodologies' critical success factors, performance expectancy (as shown in Figure 3.4) describes the degree to which an individual team member believes that using agile methodologies is critical to them accomplishing their work processes (Hardgrave, Davis & Riemenschneider, 2003; Venkatesh et al., 2003). It is the individual user's belief that using agile methodologies will assist them to achieve improvement in their job performance (Venkatesh, 2000; Venkatesh et al., 2003). The performance expectancy decision variable is associated with the following elements of the other theoretical frameworks discussed: extrinsic motivation (MM), perceived usefulness (TAM, and combined TAM-TPB), job-fit (MPCU), outcome expectancy (SCT) and relative advantage (DOI) (Davis, 1989; Moore & Benbasat, 1991). Performance expectancy represents the individual team member's perceived probability of success when using agile methodologies in software development. Perceived probability is prejudiced by their perception of the difficulty of the task, their concept of their own ability, their causal ascriptions, their perception of others' opportunities, their gender identity, their personal experiences, their emotional experiences, as well as the quality expectations, time and budget allocations and cost of failure (Montana & Charnov, 2008). Performance expectancy predicts that individual team members will be interested when they believe that the benefits they will accrue by using the system will counterbalance the disadvantages they will face when actually doing the task.

3.10.2 Effort expectancy

Effort expectancy refers to the degree to which individual team members perceive agile methodologies as easy to use (Venkatesh et al., 2003). Davis (1989) described optimal effort expectancy as being a task that is free of effort or with high perceived ease of use. Effort expectancy is interconnected to individual team members' self-assessment of how easy it will be to use agile methodologies throughout a project. It is important to know that individual team members will be interested to use an application when they perceive it as easy to use. As Bandura (1986) discovered, the easier a software application is to engage with, the greater the customer's idea of its usefulness will be. Davis (1989) concurs with Bandura (1986) and argues that instrumentality and self-efficacy are the elementary factors for perceived ease of use to influence technology acceptance. Therefore, effort expectancy and performance expectancy relate to the technological aspect of the software application in terms of its perceived complexity and its relative advantage respectively (Bandura, 1986; Davis, 1989; Venkatesh et al., 2003). In addition, Venkatesh et al. (2003) revealed that when effort expectancy is controlled by experience, gender, and age its impact on behavioural intentions is high. Moreover, they recognised that such impact can be stronger for inexperienced female customers, and experienced employees in early phases of gaining knowledge in a working environment.

3.10.3 Social influence

Social influence is the degree to which individual team members believe that other significant persons want them to use agile methodologies in the software development project (Venkatesh et al., 2003). Social influence is associated to subjective norms (TAM-TPB, TPB/DTPB, and TAM2 which adapted from TRA/TPB), social factors (MPCU), and image (DOI) (Venkatesh et al., 2003). In all of these theoretical frameworks it was recognised that social influence is important when use is obligatory and irrelevant in voluntary use.

Ajzen and Fishbein (2001) claim that, as one cooperates with other groups or individuals a person's opinions, behaviours, feelings, or attitudes have a habit of changing to orient themselves to that person's. They further note that the individual will conform to the other's behaviour or opinion in order to fit into a given condition or to get the opportunities given to another. In an organisational structure of ICT, some customer representative or users may change their behaviours and feelings towards agile methodologies, when they have a positive attitude towards it. This effect of social influence on intention is referred to as "compliance" (Warshaw, 1980).

However, as direct experience increases with knowledge over time, individuals have a better valuation of the benefits and costs linked with using that knowledge. Thus, the direct effect of social influence on intention is reduced (Warshaw, 1980). Individuals will still adopt others' ideas, particularly if they are consistent with the consequences of their own direct experience. IS research has suggested that when there is a lack of direct behavioural experience with the target object, individuals anchor their insights to abstract criteria, which in this case includes conforming with the thoughts of managers and peers. By cumulative experience, individual decisions echo actual standards flowing from communication with the target object and less from social influences. Furthermore, research has shown that the direct effect of social influences on intention is robust in the initial phases of a novel behaviour and tends to drop off over time (Reinecke, Schmidt & Ajzen, 1996). In the context of agile methodologies, this suggests that the effect of managers and peers will reduce to non-significance over time as each individual’s cumulative experience with agile increases.

In addition, UTAUT holds that social influence is affected by experience, voluntariness, age and gender. Venkatesh et al. (2003) found that female customers are more likely to be influenced by their social environments, particularly in obligatory surroundings and in circumstances when they have no experience.

3.10.4 Facilitating conditions

Triandis (1980a) defines facilitating conditions as objective factors in the environment that permit an act to be completed. In the agile context, facilitating conditions can be defined as the degree to which customers and team members believe that the software development organisation’s personnel, policies and technical capabilities exist to enable use of agile methodologies to develop a software system (Venkatesh et al., 2003). According to Venkatesh et al. (2003), facilitating conditions were discovered to have a direct effect on actual success (usage). Facilitating conditions in relation to agile development include daily meetings, fast internet connection, and easily accessible information for communication. Taylor and Todd (1995a) associated facilitating conditions to the trialability decision variable of DOI in which it was found that the occurrence of facilitating conditions do not automatically impact usage (Moore & Benbasat, 1991; Rogers & Quinlan, 2004). (All comparison of decision variables for the eighty theoretical frameworks are shown in Table 3.1) Nevertheless, Tibenderana, Ogao, Ikoja-Odongo and Wokadala (2010), found that the lack of facilitating conditions can create a barrier to usage. Together, facilitating conditions and social influence correspond with the customers’ social environment and the organisation’s internal factors.

3.11 Comparative survey of constructs in Theories of Technology Acceptance

Numerous models have been developed to inform IS research in the field of acceptance and use of technology (Venkatesh et al., 2003). The significance of each of these conceptual frameworks is relative to the research study in which it is used, the technology and the nature of the study. Several studies were designed to address the drawbacks of existing theories, others just wanted to confirm a theory‘s applicability in a different setting (Venkatesh et al., 2003). This led to theories that provide stronger and better predictions. Other research studies have tried to link diverse theoretical frameworks to come up with a new theory (Venkatesh et al., 2003). Several researchers have argued that this generally led to confusion. However, combined models have been supported by numerous researchers who have argued that it has assisted the research field to gain maturity so that the established models can be applied to a wider diversity of technologies.

As mentioned above, this study includes a comparative survey of common constructs in nine major models of innovation and technology acceptance. It is clear that several of the decision variables just change names from one model to another but are in essence the same (as shown in Table 3.1). The Unified Theory of Acceptance and Use of Technology decision variables (effort expectancy, performance expectance, social influence and facilitating condition) are compared with decision variables from the other eight models. A number of decision variables were common across the models, namely perceived behaviour control, attitude toward behaviour, results demonstrability, visibility, compatibility, and self-efficacy.

Table 3.1: A comparison of constructs used in UTAUT

UTAUT	TAM	TRA	TPB	MM	DOI	SCT	MPCU	Combined TAM-TPB
Effort Expectancy	Perceived Ease of Use				Perceived Ease of Use		Complexity Of Use	
Performance Expectancy	Perceived Usefulness			Extrinsic Motivation			Job-fit with Use	Perceived Usefulness
Social Influence	Subject Norm	Subject Norm	Subject Norm		Image		Social Factors Influences	Subject Norm
Facilitating Conditions							Facilitating Conditions	
		Attitude Toward Behaviour	Attitude Toward Behaviour			Affect	Attitude Toward Use	Attitude Toward Behaviour
		Perceived Behaviour Control			Relative Advantage	Performance Expectations		Perceived Behaviour Control
					Visibility			
					Compatibility			
					Results			
					Demonstrability			
						Self-efficacy		

3.12 Relationship between figures 2.12 and 3.4

Social Influence

Cabrera, Collins and Salgado, (2006) found that social influence had a significant impact on the success of agile software development projects. If programmers believe that they have the ability to use an agile methodology, they are more likely to use it for software development, and engage in associated knowledge management activities which may not be specified by the job requirements. Hence, programmers with high social influence will be able to take full benefit of the agile methodology to perform their tasks, and also find it easier to use.

Culture Factors

Culture is formed as users become familiar with their surroundings. Familiarising themselves with their settings, they are able to address common social challenges and adopt the essentials that lead to success (Boehm & Turner, 2005; Triandis, 1980a). Consequently, the organisational culture and the particular social community in any organisation vitally impact on users' attitudes and behaviours towards learning and regulating new inventions in the organisation (Koch, 2005; Triandis, 1980). Therefore, users within various cultural settings will have diverse behavioural intentions, meaning attitudes towards the success of agile.

Political Factors

Politics are external factors that have previously been argued to have a positive direct impact on the success of the agile software development projects (Koch, 2005; Venkatesh et al., 2003). For example, if agile professionals continuously support junior software developers and train them, the organisation will in turn assist other junior software developers to attain self-efficacy with the scheme. Furthermore, it is presumed that political impacts are entrenched in the users' day to day events. That means that social events are not only affected by the people they perceive to be significant in their specific job activities, but moreover by their environments (Venkatesh et al., 2003).

Organisation Factors

Numerous researchers, such as DeLone and McLean (2002) and Curtis and Payne (2008), suggest that organisational factors play an important role in the use of agile methodologies which ultimately leads to the success of agile software development projects. Similarly, research on agile methodologies showed that some of the top critical success factors of agile methodologies are organisational factors (Koch, 2005; DeLone & McLean, 2002). Therefore, scientists have found that top executive or management support is one of the first priorities with regards to agile methodologies (Chow & Cao, 2008).

People Factors

According to Triandis (1980b), people's beliefs play a significant part in whether users believe a system is easy or difficult to use. This explains why, when people believe that a system is easy to implement, they will implement it, and if not why they will avoid it (DeLone & McLean, 2002). So when users perceive a system as easy to use and they go ahead and use it. This means that it is also easier for them to understand its benefits. Likewise, users' viewpoints have an effect on their societal beliefs. Bossini and Fernández (2013), and Boehm and Turner (2003c), have noted that, individual or people factors are critical success factors affecting software projects using agile methodologies.

Technology Factors

This research shows that technological factors can have direct influence on the success of the agile software development project respectively (Marnewick & Labuschagne, 2009; Augustine, Misra et al., 2009; Ceschi, Sillitti, Succi & De Panfilis, 2005, Venkatesh et al., 2003).

Process Factors

This research shows that process factors can have direct influence on the success of the agile software development project respectively (Marnewick & Labuschagne, 2009; Augustine, Misra et al., 2009; Ceschi, Sillitti, Succi & De Panfilis, 2005, Venkatesh et al., 2003).

Project Factors

This research shows that, project factors can have direct influence on the success of the agile software development project respectively (Marnewick & Labuschagne, 2009; Augustine, Misra et al., 2009; Ceschi, Sillitti, Succi & De Panfilis, 2005, Venkatesh et al., 2003).

Vendor Factors

This research shows that, vendor factors can have direct influence on the success of the agile software development project respectively (Marnewick & Labuschagne, 2009; Augustine, Misra et al., 2009; Ceschi, Sillitti, Succi & De Panfilis, 2005, Venkatesh et al., 2003).

Performance expectancy

The factors that affect agile software development projects include perceived usefulness, and perceived ease of use. Perceived usefulness and perceived ease of use have been discovered to affect user perceptions of different technology innovations, such as mobile commerce and agile software development projects (Venkatesh et al, 2003; Chan & Thong, 2009).

Effort expectancy

Perceived ease of use is defined as the degree to which an individual believes that using a particular agile methodology would be free of physical and mental effort (Moore and Benbasat, 1991). Venkatesh et al. (2003) regarded perceived ease of use as effort expectancy. Effort expectancy has been found to be an important factor influencing the intention to adopt an innovation in previous studies on several technologies (Davis, 1989).

Intention Factors

The dependent variable in this study is agile professionals' intention to use agile methodologies (Hardgrave et al., 2003). Intention is normally used as an indicator of the use and acceptance of technology adoption in UTAUT (Venkatesh et al., 2003). Intention has become the de facto measure for evaluating the acceptance of an invention and has continually proven to be a strong predictor of actual use (in other words, the success of software projects) (Venkatesh et al., 2003). According to Ajzen (1991), intentions are expected to capture the motivational factors that have influence on a behaviour; they are signs of how hard people are willing to try, of how much of an effort they are planning to exert in order to perform the behaviour, based on past experience.

3.13 Summary

In this chapter, the theoretical framework that forms the basis for this study was discussed. The major aim of this study is to establish the critical success factors that influence the effective use of agile methodologies of software development. This chapter discussed the literature of technology acceptance and use. Greater emphasis was placed on the UTAUT model, while a brief discussion was set out of the eight other models that were combined to develop UTAUT. Based on the recommendations and literature on the studies that have replicated, extended and validated UTAUT, this study has found UTAUT to be the most suitable model for validating critical success factors in order to come up with an appropriate conceptual framework for acceptance and use of agile methodologies.

CHAPTER 4: RESEARCH DESIGN AND METHODOLOGY

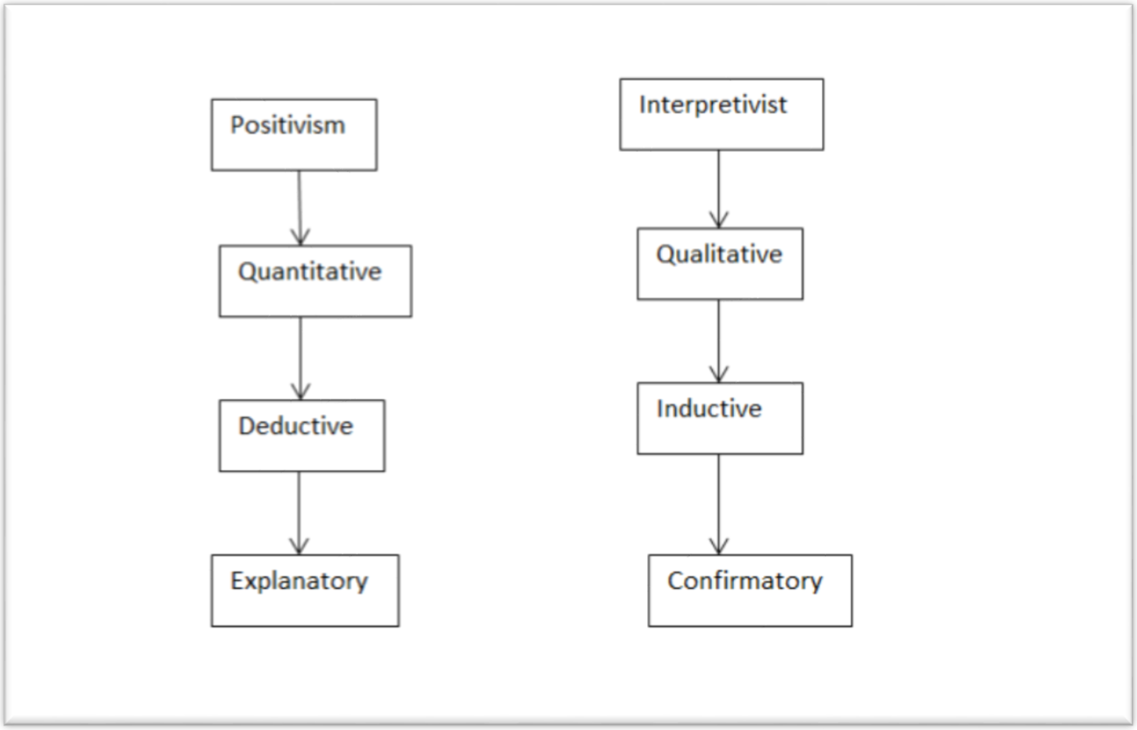
4.1 General research approaches and philosophies

There are a number of dichotomous paradigms and perspectives that have underlined research approaches in science, such as interpretivist and positivist research, qualitative and quantitative research, deductive and inductive research, and confirmatory and exploratory research (Creswell, Hanson, Plano, & Morales, 2007). A paradigm is a set of assumptions that provides a philosophical world view or conceptual framework which enables organised study of the specific population (De Villiers, 2012). A paradigm serves the following purposes: (1) it establishes standards for tools such as instruments, methodology and data collection that would enable researching the matter, (2) it provides the procedures, philosophies and methods to be considered when similar concerns appear again (3) it guides professionals as it specifies important matters challenging any discipline; and (4) it develops theories and models that permit specialists to solve matters (Creswell, 2013).

At the same time that philosophers were answering the fundamental questions: how do you know for certain that what we know is true? (Creswell, 2014), other researchers – such as Creswell et al. (2007) and Bryman (2015) – were addressing the epistemological side of paradigm development and different schools of thought adopted the apparently dichotomous views of positivism and interpretivism (Bryman, 2015). (Interpretivism has numerous names, including naturalism or constructivism (Creswell, 2014)). Researchers have traditionally distinguished positivism and interpretivism in a way that makes them mutually exclusive or dichotomous. However, many researchers now advocate more practical approaches that do not emphasise such differences and clearly recognise common beliefs that exist in both approaches, hence, the call for a complementary approach that make the most of both approaches (Hayes, 2000a; Johnson, Onwuegbuzie & Turner, 2016).

The logical link between sections 4.1.1 up to 4.1.4 is shown in figure 4.1a. The research was a mixed method which follows positivism, quantitative, deductive and explanatory mostly. Also focus on qualitative (interpretivist, inductive and confirmatory) on a lesser extend compared quantitative as shown in figure 4.1a.

Figure 4.1a: Logical links of the research concepts



4.1.1 Positivism versus interpretivism

The underlying assumptions of qualitative research are based on specific research paradigms, the basic ones being positivism and interpretivism (De Villiers, 2012; Nardi, 2015). The research paradigm is a vital aspect of the research process, and Remenyi and Williams (1998) note that the basic beliefs that define a specific research paradigm can be summarised by the responses given to three major questions: (1) the ontological question such as, what is the form and nature of reality?, (2) the epistemological question, such as, what is the basic belief about knowledge? and (3) the methodological question, such as, how can the researcher go about finding out whatever he or she believes can be known?

Ontology refers to the nature of social reality and epistemology refers to the nature of knowing and the construction of knowledge (Remenyi & Williams, 1998). Creswell et al. (2007) and Johnson et al. (2016) remind researchers that they must initiate their studies with an analysis and interpretation of the philosophical perspective, questioning what is known and how they know it (epistemology), the nature of reality (ontology), the dependence or independence of the subject (role of researcher), and the nature of the emergence of the research (methodology).

In order to decide the most appropriate paradigm for this study two common classifications offered by researchers and scholars were identified: interpretivism and positivism as shown in Table 4.1 (Creswell, 2013).

Table 4.1: Research paradigm (Creswell, 2013)

Research paradigm		
Research paradigms	**Positivist**	**Interpretivist**
Ontological	There is an objective world in which true reality exists with stable pre-existing patterns, and which science can mirror with privileged knowledge and understanding. (Realism)	There is a complex and dynamic world which is interpreted and experienced by human beings, meaning that reality is socially constructed. (Nominalism)
Epistemological	Reality can be verified probabilistically and hypothetically, and knowledge is certain and accurate. (Positivism)	Knowledge is gathered through subjective behaviours and beliefs, and observed phenomena. In this way people make meaning in context. (Anti-positivism)
Role of researcher	The researcher is objective and independent of the subject, and values have no place in research. This necessarily eliminates all bias. (Determinism)	The researcher brings their own subjective experience to the research, and recognises that values are an integral part of social life. (Voluntarism)
Methods	Research is structured and empirical and methods include experiment, observation, surveys, verification of hypothesis, statistical analysis, quantitative descriptive studies, tests and scales. (Nomothetic)	Research methods include unstructured interviews, observation, open field research conducted in natural settings, action research, ethnography, participant observation, case studies, in depth surveys etc. (Ideographic)

Creswell et al. (2007) have argued that the so-called meta theoretical differences between interpretivism and positivism are specious. Creswell (2013) suggested that the differences essentially depend on which methods have been elected; a positivist researcher could use research methods such as surveys, experiments and field studies, whereas interpretivist researchers could make use of ethnomethological studies, phenomenographic studies, ethnographic studies and case studies. Researchers have different ways of selecting research methods because of a number of factors, for example the type of training offered for the researcher, societal pressures related to colleagues and consultants, and which methods are likely to result favourably in receiving particular types of insight during the research being carried out. Creswell et al. (2007) recommended that it is time to place the rhetoric of interpretivism against positivism to rest, as it serves no useful purpose but instead promotes prejudgment in research assessment (Nasehi, 2013). Furthermore, he adds that the researcher’s objectives are to improve knowledge of a particular phenomenon, recognising that different research methods and data analysis techniques have their own unique advantages and disadvantages, based on existing knowledge pertaining to the phenomena (Nasehi, 2013). Table 4.1 shows possible research methods and discusses advantages and disadvantages for each method. According to Creswell (2013), field experiments, formal theorem proof, simulation, surveys and case studies are some of the research methods accepted within positivist-based research, whereas action research, case studies, ethnographic studies and meta-analysis is accepted within interpretivist-based research.

Table 4.2: Summary of methodological dichotomies

Qualitative: A qualitative methodology is a contextually-based study of socio-technical environments where reality is perceived as a composite of multiple and subjective views (Klassen, Creswell, Plano Clark, Smith & Meissner, 2012). This approach can also be defined as socio-technical to some extent (Maxwell, 2012).	**Quantitative:** Bavelas (1995) describes quantitative research as a method for testing theories by studying and investigate the relationships among constructs. The constructs can be calculated, usually by means of a particular measuring tool, so that numerical data can be analysed using statistical methods.
Induction: Inductive research involves the study pattern from observation and the development of descriptions of the theories with hypotheses. Theories or hypotheses are applied at the last stage and the researcher is free in terms of changing the direction for the research after the research process had begun (Eriksson & Kovalainen, 2015).	**Deduction:** Deductive reasoning is a logical process in which a conclusion relies on the concordance of multiple premises that are normally assumed to be true (Nardi, 2015).
Exploratory: Concerned with discovering patterns in research data in order to understand and examine of the phenomenon (Moen & Middelthon, 2015).	**Confirmatory:** Concerned with hypothesis testing and verification theory, thereby following positivist, quantitative methods of research (Fink, 2015).

Investigator triangulation consists of using more than one field researcher to collect and analyse the data relevant to a specific research study (Leedy & Ormrod, 2005). Methodological triangulation involves the combination of different research methods and the theory of triangulation and making explicit reference to more than one theoretical tradition to analyse data

(Kolb, 2008). De Vaus and de Vaus (2001) clearly defined two forms of methodological triangulation: within-methods triangulation which involves making use of different varieties of the same method, and between-methods triangulation which involves making use of different methods, such as qualitative and quantitative methods, in combination. Creswell (2013) identified methods in which qualitative and qualitative methods can be combined efficiently and effectively in different phases of the research process, as described and shown in Table 4.3. In addition Creswell (2013) suggested three methodological paradigms, also shown in Table 4.3.

Table 4.3: Types of Research

Approaches	Key features	Strengths	Weaknesses
Experimental Research **(Keppel, 1991)**	Seeks to determine if a specific action influences the results. This influence is evaluated by providing a specific action to one group and withholding it from another and then determining how both groups score on the results.	1. This research helps in controlling independent variables of the experiments and aims to remove unnecessary variables. The control over the unrelated variables is higher as compared to other research methods. 2. The experimental design of this type of research approach consists of manipulating independent variables to easily determine the cause and effect of the relationship.	1. Just like any other type of research, experimental research is subject to human error and this will someway affect the efficiency of the outcomes.
Survey Research **(Babbie, 1990)**	Provides a quantitative or numeric explanation of trends and opinions of a population by studying a sample of that population. It comprises longitudinal and cross-sectional studies through use of structured interviews or questionnaires for data collection, with the idea of generalising from a sample to a population regarding relations that exist in past, present and future.	1. Can be developed in less time in comparison to the other data collection methods and is cost-effective, but cost relies on the survey mode. 2. Can be controlled remotely via online, mobile devices, mail, email or telephone. 3. Enables collection of data from a large number of respondents. 4. Several questions can be asked about a topic, giving extensive flexibility in data analysis. 5. With survey software, advanced statistical techniques can be used to analyse survey data to find out validity, reliability, and statistical significance, including	1. Questions that tolerate controversies might not be exactly responded to by the respondents because of the probable difficulty of recalling the information associated with them. 2. The survey that was utilised by the researcher from the start, as well as the method of administering it, cannot be altered during the process of data gathering.

Approaches	Key features	Strengths	Weaknesses
		the ability to analyse multiple variables. 6. A broad range of data can be collected such as attitudes, opinions, beliefs, behaviour, values and factual.	
Case studies (Descriptive research) (De Vaus & de Vaus, 2001)	A strategy of inquiry in which the researcher discovers in depth a program activity, event, process on one or more group of organisations. Cases are restricted by activity and time, and researchers collect detailed information using a variety of data collection steps over a persistent period of time. The primary source of data is interviews complemented by documentary evidence.	1. Case studies permit a lot of aspects to be collected that would not normally be easily attained by other research designs. The data collected is usually a lot richer and of greater depth than can be found through other experimental designs. 2. Case studies can enable pseudoscientists to adapt ideas and produce new hypotheses which can be utilised for later testing.	1. One of the major criticisms is that the data collected cannot be generalised to the broader population. Therefore, data being collected over longitudinal case studies is not always relevant or particularly useful.
Ethnography research (Babbie, 2015)	Strategy of inquiry in which the researcher studies a whole cultural group in a natural setting over a lengthy period of time by primarily collecting observation and interview data. The research process is flexible and normally evolves contextually in response to the lived realities encountered in the field setting.	1. Direct observation: The researcher is not relying on second hand reporting, but is able to collect data that he or she has observed at first hand, and hence knows that there have been no errors. 2. Links with a theory: The researcher is evaluating the material collected and can compare this with the theories,	1. Needs a vast investment of the researcher's time, hence some studies can go on for years, and the researcher is required to be part of the group for all that time. 2. Organising all the data and results into a coherent presentation or paper can be very difficult.

Approaches	Key features	Strengths	Weaknesses
		changing the theories as the data commands. 3. Detailed data: As for the length of time spent with the participants, as well as the close proximity and the researcher can obtain relevant data indeed through observations.	
Phenomenological research (Moustakas, 1994)	Strategy of inquiry in which the researcher recognises the importance of human experiences as they pertain to a phenomenon as explained by respondents. Understanding the lived experiences marks phenomenology as a philosophy as well as a method, and the procedure includes studying a small number of subjects through extensive and prolonged engagement to develop relationships and patterns of meaning. Hereby, the researcher sets aside her or his own experiences in order to understand the respondents in the study through interviews etc.	1. Phenomenological research creates understanding of meanings attached by people and its contribution to the development of new theories. 2. Data is gathered which is seen as natural rather than artificial.	1. Phenomenological research faces difficulties with analysis and interpretation, and generally has lower levels of validity and reliability compared to positivism. 2. Phenomenological research needs more time and several resources required for data collection. 3. Participants must be interested and articulate problems, this can cause difficulties if they are unable to express themselves well.
Ethnomethodological studies (Moustakas,1994)	Empirical study of the methods that people use, and the focus of the perspective is on the practices through which members construct their social world.	1. Practical orientation: encourages being responsive to situations of interest to the experts.	1. Does not formulate a theoretical basis for understanding events.

Approaches	Key features	Strengths	Weaknesses
Action Research **(Moustakas, 1994)**	Action research is a disciplined process of inquiry carried out by the researcher and those taking the action. The primary reason for engaging in action research is to help the "actor" in improving or refining his or her actions.	1. Practical and theoretical emancipated paybacks for both researcher and researched organisation are achieved. 2. The biases of researcher are made known 3. Action research is a flexible spiral process which allows change, improvement and understanding knowledge to be achieved at the same time.	1. Involves studying different societal cultures, wider than a single organisation. 2. Involves complex issues in different cultures thus the variables are too vast. 3. Action research requires implementation of the findings to help proceed to the next stage of action research.

4.1.2 Quantitative versus qualitative

Quantitative empirical research uses administered surveys, laboratory based experiments, quantitative metrics, highly structured protocol simulations and hypothesis testing to build a body of knowledge around a particular area of study (Olszewska, Heidenberg, Weijola, Mikkonen & Porres, 2016). The use of quantitative empirical studies is well developed in the natural sciences research setting where indicators that represent the truth have been developed (Henning et al., 2004). Furthermore, quantitative researchers conduct the inquiry in an unbiased, objective manner, through the description of trends or an explanation of the constructs' relationship regarded as positivistic (De Villiers, 2012; Nardi, 2015). Since quantitative research supports a positivist epistemological perspective, the researcher and research object examined are regarded as independent objects, in the sense that the researcher is able to study the occurrences without influencing them or being influenced by the environment.

Qualitative data is usually collected in the form of words or images rather than numbers and has always been the staple of some fields in the social sciences, notably, history, anthropology and political science (Henning et al., 2004). Qualitative research methods were developed in the social sciences to enable researchers to study human behaviour and belief phenomena (Marshall & Rossman, 2014). These were designed to help researchers understand people and the social and cultural contexts within which they live. Moreover, the qualitative approach permits a further definition of the study's nature and restrictions (Knox, 2004). According to Henning et al. (2004) a socio-technical perspective looks at people and technical features, how they are used, and how they interact. Since the research supports an interpretivist epistemological perspective, the researcher and research object examined are interactively linked, in the sense that the researcher is able to study the occurrences while being influenced by the environment.

In conclusion, the qualitative nature of the research will help to expose hidden and unsuspected issues to be analysed (Fink, 2015). It also assists in exploring attitudes, sensitive issues, emotions, conceptions and opinions (Eriksson & Kovalainen, 2015). Furthermore, it explores context, relationships, and processes where possible. Qualitative research performed in natural settings focuses on context, is emergent and evolving, and is fundamentally interpretive (Marshall & Rossman, 2014). Qualitative researchers believe that humans are conscious of their own behaviour and beliefs, and of the thoughts, perceptions and feelings of their informants (De Villiers, 2012). Consequently, qualitative research has helped researchers to understand the social and cultural contexts of people and, in turn, to answer research questions more meaningfully. Therefore, quantitative research is utilised to answer questions about relationships among measured variables with the aim or purpose of predicting, explaining and controlling the real situation of the study (Leedy & Ormrod, 2005).

Mixed methods research is the type of research in which a researcher combines qualitative and quantitative research methods such as the use of qualitative and quantitative views, data collection, inference techniques and analyses for the broad purposes of depth of understanding and corroboration of the study (Creswell, 2013).

Kaplan and Duchon (1988) claim that combining quantitative and qualitative methodologies can be done without violating basic paradigmatic assumptions. Leedy and Ormrod (2005) recommended a matrix method for connecting quantitative and qualitative research on the data collection level, where the grouping is centred on two types of decisions which are sequenced and prioritised. According to Leedy and Ormrod's (2005) guidelines, the principal method is quantitative, but the use of a qualitative approach at the beginning is used to improve the effectiveness of the quantitative research methods that will be used. The qualitative approach should be used to check and develop the content of the quantitative questionnaire in order to confirm that the survey covers the important topics suitably.

Creswell (2013) expressed a different viewpoint on how the two approaches can complement each other at different phases within the research process. At the research design stage, quantitative data can help qualitative components, identifying members of a representative sample and spotting outlying observations (Johnson et al., 2016). Equally, qualitative data can help quantitative components with concept as well as instrument development (Creswell & Clark, 2007). At the data collection stage, quantitative data can provide baseline information and help avoid elite bias, whereas qualitative data can help facilitate the assessment of the generalizability of quantitative data and give a new perspective on the findings (Bryman, 2015). Finally, at the data analysis stage, qualitative data can play an important role in interpreting, clarifying, describing and validating quantitative results, in addition to ground and modify the theoretical perspective (Babbie, Wagner III, & Zaino, 2015).

According to Johnson et al. (2016) different types of mixed methods research can be represented on the qualitative-quantitative continuum as shown in Figure 4.1.

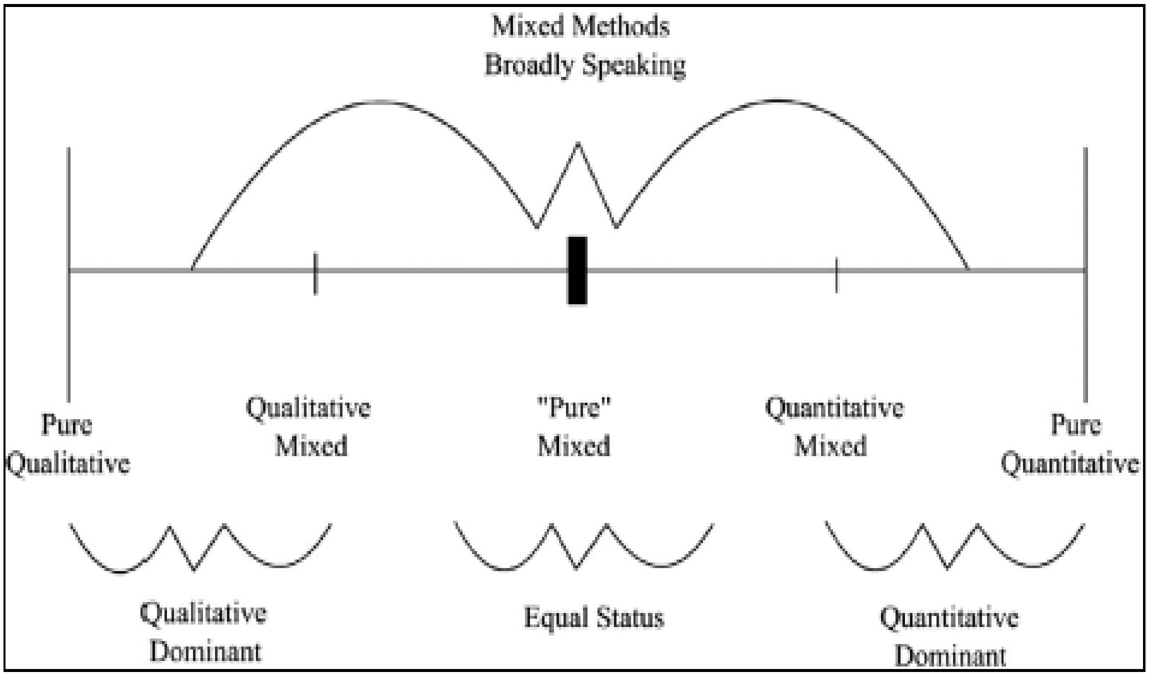


Figure 4.1: Graphic of the three major research paradigms, including subtypes of mixed methods research (Johnson, Onwuegbuzie & Turner, 2007)

Figure 4.1 illustrates the pure research, pure qualitative, and pure quantitative paradigms. The area in between the pure research extremes signifies the different groupings of mixed research paradigms (Johnson et al., 2016). A researcher that self-identifies as a mixed methods researcher, broadly speaking, falls within the centre, representing the strongest or pure form (Johnson et al., 2007). Mixed methods research that is principally qualitative (QUAL + quan) is suitable for investigators who rely on a qualitative view of the research process but want to include quantitative methods to less extent since these are likely to be an advantage most research studies (Johnson et al., 2007). Mixed methods research that is principally quantitative (QUAN + qual) favours a researcher who believes in a quantitative view of the research process and also believes it is vital to include qualitative methods and data to add value to research studies.

Mixed methods research approaches, combining qualitative and quantitative methods, as described are but one way of addressing the mix or the combination of qualitative and quantitative methods.

This research design phase is a typological process that encourages pragmatic, logical, linear flow and focuses on a number of issues: the balance of qualitative and quantitative (QUAN + QUAL) methods used in the research design, deciding on the primary method and the sequence of the research (Johnson et al., 2007). De Villiers (2012) suggested that research designs should also consider the purpose of mixing methods in the implementation phase. On a different note, Grinnell and Unrau (2005) recommended a research design approach to mixed methods that differs from typologies in more than one way; the interactive design of a study and the conceptual analysis of fundamental differences between the qualitative-quantitative approaches (Johnson et al., 2007). Creswell (2013) understands design of a study as laying out five key points: the research aim, the research questions, the conceptual framework, the methods, and the reliability and validity approaches. Johnson et al. (2007) discuss this design concept as a systematic model, since the elements are interlinked in a web rather than being linked in a cycle sequence. The five components can affect each other and the research questions play the vital role (Johnson et al., 2016). Research questions are regarded as the pillar-stone for guiding the other four components in the interactive model design (Johnson et al., 2007). Therefore, research questions are required to inform and be responsive to all the other elements of the design. Interactive design mainly involves the relationships between quantitative and qualitative methods, both across and within the five elements.

Conceptual analysis differentiates between two paradigms to examine process model and variance model (Creswell & Clark, 2007). Variance theory relies on correlations among variables and includes defined correlations between variables and measurement of differences; thus, it lends itself to research that uses extensive pre-structuring of the research, quantitative measuring, probability sampling, statistical testing of hypotheses and correlational design (Creswell, 2013; Johnson et al., 2016). Process theory refers to the events and processes that are linked based on an analysis of the causal processes by which events affect each other (Hayes, 2000b). Thus, the process theory is less well-suited to quantitative and involved in-depth study of a small number of individuals and textual forms of data that retain the contextual link between events (Keppel, 1991). Knox (2004) viewed the qualitative-quantitative (QUAN + QUAL) paradigm as grounded in the distinction between two contrasting approaches and the explanation of variance theory and process theory. The interactive and systematic approach is understood as complementary in the provision of insights and tools in the research.

Table 4.4 below shows various data collection techniques used in quantitative and qualitative mixed methods and examines the advantages and drawbacks of each approach (Chaleunvong, 2009). What follows here is a detailed description of two data collection techniques which are potentially useful in this study.

1. Structured interviews are a data collection techniques which focus on finding answers to carefully worded questions. Interviewers are educated to diverge only slightly from the question phrasing to ensure uniformity of interview administration (Balnaves & Caputi, 2001). The communication tools of structured interviews can include telephone interviews, face-to-face interviews, or interviews conducted through a medium such as internet or cell phone (Pearl, 2000). Each means of approaching the interview has its strength and drawbacks in terms of time, clarity, cost, interviewer training and knowledge of computers.

2. Questionnaires are an efficient data collection technique when researchers know exactly what is needed and how to measure the variables concerned (Hayes, 2000b). Questionnaires include groups of questions which can be administered personally in a written format, distributed electronically, or emailed to the respondents. Questionnaires are quick and easy to administer (Babbie, 1990) but very careful attention is required to ensure the wording of the questions, the layout of the forms, and the ordering of the questions ensures a valid outcome (Creswell, 2013).

This research follows both a qualitative and quantitative methodology, which is appropriate to the ‘how’ and ‘why’ types of research question articulated below (Creswell, 2013; Henning, Van Rensburg, & Smit, 2004). Although, the research is mostly quantitative, it is evident that it would benefit from a combination of approaches (Creswell, 2013). This study uses qualitative methods to discover patterns and develop theories to gain a better understanding of the subject under investigation (De Villiers, 2012). The study thus aims to generate knowledge of human action in context, through the use of qualitative data (Johnson et al., 2016).

The study largely used online questionnaires to qualitatively and quantitatively test what the main success factors are for agile software development projects. Conducting the research in different provinces in South Africa provided a better understanding of the cultural and social context of the software development community. The research provides a clear indication of the changes needed to implement agile methodologies as success factors of software projects.

Table 4.4: Data collection techniques

Collection Technique	Description	Strengths	Weakness
Observation (Creswell & Clark, 2007) (Qualitative study)	Observation is a technique that involves systematically selecting, watching and recording the behaviour and characteristics of phenomena, objects, or living beings.	1. It provides direct, detailed and context-related information. 2. It enables the collection of information on facts, which are generally not mentioned in an interview. 3. It enables testing of reliability of responses to questionnaires.	1. Ethical issues concerning confidentiality or privacy may arise. 2. Observer bias or perception may be present, meaning that the observer may only notice what interests him or her. 3. The presence of the data collector can influence the situation observed. 4. Well qualified research assistants or thorough training is required.
In-depth interviews (Kalogiratou, Monovasilis, Ramos & Simos, 2016) (Qualitative study)	This involves both individual interviews as well as group interviews.	1. It commonly yields the richest data, facts and new insights 2. It enable face-to-face contact with participants. 3. It provides a chance to explore subjects in depth. 4. It allow the interviewer to describe or clarify questions, helping to ensure that valuable responses are elicited.	1. It is time consuming and expensive. 2. There need to be well-qualified and highly trained interviewers. 3. The interviewee might alter the information they provide as a result of recall error and a desire to please the interviewer 4. Flexibility can lead in discrepancies across interviews. **5.** Poor quality audio recordings and stenography might be difficult to transcribe.

Collection Technique	Description	Strengths	Weakness
Focus Group (Carey & Asbury, 2016) (Qualitative study)	Focus group discussion permits a group of 8 –12 informants to freely discuss a certain subject with the guidance of a facilitator through interviewing and participant observation.	1. It is quick and relatively easy to develop relevant research hypotheses by using focus groups to explore in greater depth the problem to be investigated and its possible causes. 2. Focus groups are useful for formulating appropriate questions for more structured, or larger scale surveys. 3. They enable the researcher to understand and solve unexpected problems in interventions. 4. Focus groups can be used to develop suitable messages in software projects and also to later evaluate the message for clarity. **5.** They are good for exploring controversial topics.	1. Discussion can be controlled by a few people. 2. Data analysis is time consuming and must be well planned. 3. Data is not always easily accessible. 4. Ethical issues concerning confidentiality may arise. 5. The information provided by different groups may be imprecise or incomplete.
Administering written or online questionnaires (Creswell, 2013) (Quatitative study)	The participants are given questionnaires to complete in their own time and space. The most common form of self-administered questionnaires are email surveys.	1. This is typically less expensive than other techniques. 2. It permits anonymity and may result in more honest responses. 3. It does not need research assistants. 4. It eliminates bias which can be introduced when questions are phrased differently with different respondents.	1. It cannot be used by illiterate respondents. 2. It require some extra training of researchers.

4.1.3 Deductive versus inductive

The inductive and deductive methods are two comprehensive approaches of reasoning (Gioia, Corley & Hamilton, 2013). Inductive reasoning is a process of reasoning in which if the evidence supporting an argument is believed it supports the conclusion but does not ensure it (De Villiers, 2012). In addition, researchers logically establish a general proposition based on observed phenomena (De Vaus & de Vaus, 2001). Inductive reasoning is more exploratory and open-ended than deductive reasoning (Hayes, 2000b). Inductive reasoning begins with precise observations and results in wider generalised theories using a bottom-up approach (bottom-up logic), which is useful for detecting regularities and patterns, which can lead to the formulation of some tentative hypotheses that can be explored, and which ultimately provides strong evidence to support a conclusion (Henning et al., 2004). Creswell (2014) recommended the use of an inductive approach if there is not enough pre-existing knowledge in the area of the study.

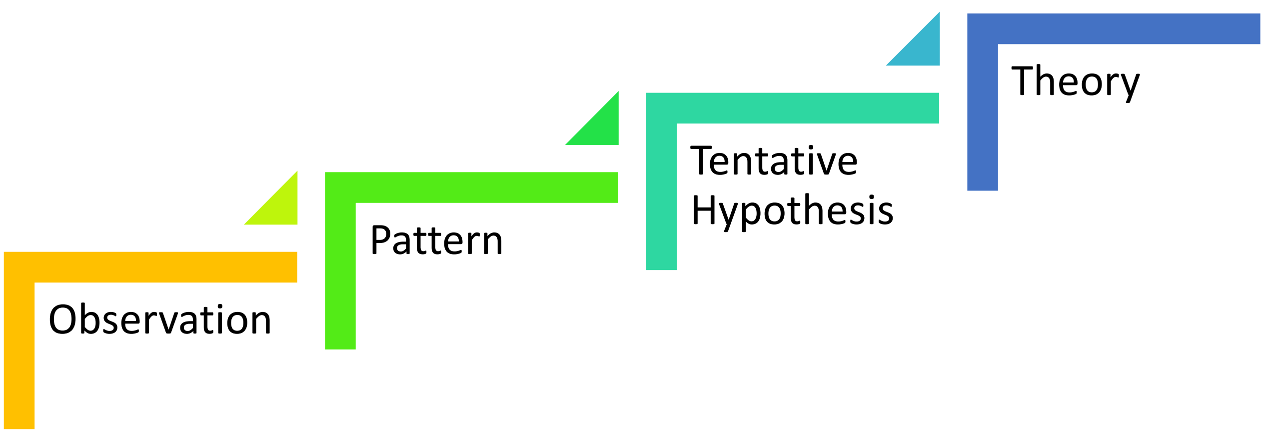


Figure 4.2: Inductive reasoning diagram (Rocco, Bliss, Gallagher & Pérez-Prado, 2003)

Deductive reasoning is narrower in nature and is concerned with confirming hypotheses (Kalogiratou et al., 2016). It involves a set of techniques for applying rigorously testable theories in the real world in order to assess their validity (Olszewska et al., 2016). Deductive reasoning moves from the wide ranging to the specific, in a top-down manner (top-down logic) (De Vaus & de Vaus, 2001). This process involves a number of steps that are regarded as the building blocks of the scientific method (Creswell, 2013). The first stage in the deductive process is the generation of theories and then hypothesis (Creswell, 2013). The generation of ideas is centred on individual experiences and theories, and then hypotheses that arise from a literature search of the specific study (De Villiers, 2012). The second stage is operationalising the concepts in the hypotheses and theories in such a way that those ideas can be statistically evaluated through empirical observations (Boehm et al., 1976). The succeeding stage in the process comprises deciding and identifying between alternative approaches for measuring the operationalised ideas (Henning et al., 2004). This includes the research design and selection of research methodologies to be used, such as a research instrument, sampling plan, data collection approaches, and approaches of analysis and interpretation of empirical measurements and observations (Creswell, 2013).

Creswell (2013) supported using both inductive exploratory questions and deductive confirmatory questions in the same study, holding that qualitative questions are mainly inductive questions, while quantitative questions are deductive hypothesis testing questions. Therefore, in mixed methods research, the quantitative or qualitative elements include the essentials of both deductive and inductive influences (Knox, 2004). Thus in mixed methods research, deduction supplies the shape of the argument and induction establishes agreement about one or more pieces of the argument (Henning et al., 2004). The two approaches of reasoning are connected in the observation phase: the investigator observes patterns in the data that guide him to develop new hypotheses and theories (Creswell et al., 2007). Hence, deductive and inductive reasoning are associated: inductive is used to show that a causal relationship exists and to build facts on which the deduction is formulated (Creswell, 2013).

As a largely quantitative study, deductive reasoning is mostly used as the research seeks to validate or disprove existing inquiries into critical success factors in the context of agile software development.

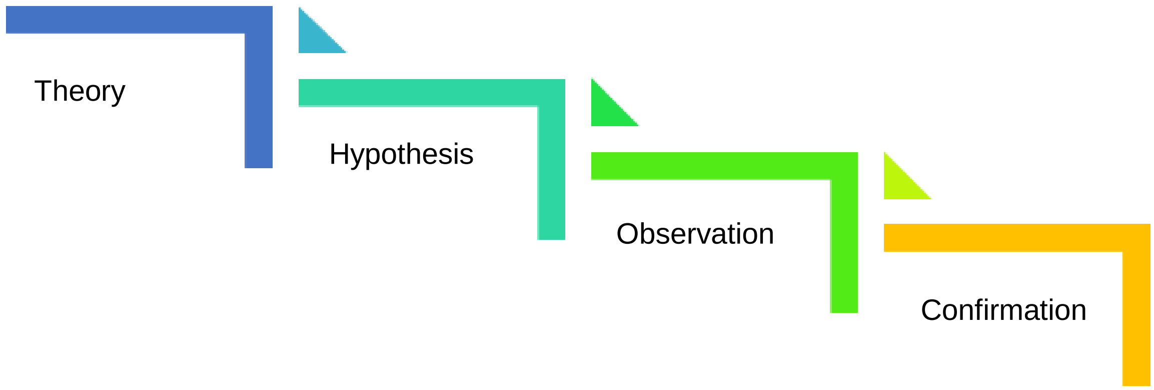


Figure 4.3: Deductive reasoning diagram (Rocco et al., 2003)

4.1.4 Exploratory versus confirmatory

Exploratory and confirmatory approaches are the two most common research strategies (Venkatesh, Brown & Bala, 2013). Creswell (2013) and Bavelas (1995) note that confirmatory studies are those where the researchers are seeking to confirm a pre-defined, specified relationship, whilst exploratory research strategies are used when researchers are interested in clarifying the most common relationships and then using multivariate techniques to estimate any casual effects.

Accordingly, a confirmatory approach of study (confirmatory research strategy) is defined as the strategy that uses empirical analysis to disapprove or confirm the proposed hypotheses (Bavelas, 1995; Mugenda, 1999). On the other hand, an exploratory study approach focuses on the evidence and theory which are closely related (Nasehi, 2013). The hypothesis were formulated from previous studies and verified with interview. Therefore, the approaches used in the design phase of confirmatory research and exploratory research are dichotomous. An exploratory approach to a study is mostly agreed as a process of common amendment whereby the ideas and the theories correctly group together (Punch, 2013). The confirmatory approach to a study is commonly supported by researchers with hypothetical and experimental backgrounds, while an exploratory approach of the study is commonly supported by those with an interpretivist alignment (Creswell, 2013; Olszewska et al., 2016). Researchers getting used to the exploratory approach of study commonly leave matters of theorisation, conceptualisation, and inquiry open so that they are able to be thoughtful to the suggestions at hand.

Exploratory research must be carried out in a transparent way, with more realistic statements of accuracy, flexible ways to generate hypothesis, promotion of deeper understanding of processes and a trail of set procedures that ensure its reliability (Kaplan & Duchon, 1988; Venkatesh et al., 2013). Exploratory research, if carried out using appropriate guidelines, can achieve appropriate validity and it can offer new ways to analyse the reality. The method of exploratory research can result in problematic theory distortion where outcomes are over-stated with a higher possibility of wrong results (Punch, 2013). Exploratory research is regarded as inductive in nature.

Confirmatory research is based on arithmetical inferences and the deductive approach of the descriptive statistics (De Villiers, 2012). The hypotheses are defined and then verified by answering particular questions (Creswell & Clark, 2007). This shows the benefits of the confirmatory analysis of providing exact information in the right circumstances whilst using well recognised methods and theory. However, the drawbacks of the confirmatory research approach lie in the misleading impression of precision it can create in less than ideal circumstances, the analysis driven by preconceived ideas and the difficulty of noticing unexpected results. Most researchers have used confirmatory approaches during their degree studies therefore this is the usually supported model of analysis.

Creswell (2013) suggested that most social science research falls between confirmatory and exploratory principles. Creswell et al. (2007) claim that confirmatory and exploratory approaches of reasoning in social science are possibly more alike than dissimilar, despite the entire difference in their aims. According to Babbie (2015) some divergence between the two methods of reasoning exists in terms of their aims. In this view, exploratory research aims to create novel ideas and combine them, relying heavily on probability models that develop directly from the data. On the other hand, confirmatory research aims to evaluate hypotheses and check the validity of the assumptions in the research design. Exploratory research is looking for flexible ways to examine data without preconception, relies heavily on graphical displays, lets data suggest questions, focuses on indicators and approximate error magnitudes and requires the researcher to be open minded. Confirmatory research is based on hypothesis tests and formal confidence interval estimation. The hypotheses determine how the data will be collected, there is emphasis on numerical calculations, researchers look for definite answers to specific questions, and hypotheses are used to control variables and predict results (De Vaus & de Vaus, 2001). According to Creswell (2013) both confirmatory and exploratory research can be either quantitative or qualitative.

Babbie (1990) explained how the phases in the process of research reflect confirmatory or explorative strategies in both quantitative and qualitative research. The first phase (in which data questions are developed and the research problem is outlined) is a confirmatory process within the quantitative approach, while in the qualitative approach it is an exploratory process which focuses on descriptive statistics (Bagozzi & Edwards, 1998; Mugenda, 1999; Ritchie, Lewis, Nicholls & Ormston, 2013). In the second phase (in which data is collected) quantitative confirmatory research employs instruments, observation, score-oriented closed-ended process and proposed hypotheses, while qualitative exploratory research can include interviews, observation, open-ended process and video-oriented approaches (Kolb, 2008; Nasehi, 2013). In the data analysis phase, quantitative confirmatory research relies on measures such as descriptive statistics and inferential statistics, while qualitative exploratory research uses procedures such as descriptive statistics (including classifying themes and looking for associations among themes (De Villiers, 2012; Nasehi, 2013). Data interpretation is the final phase in the research process and quantitative confirmatory research focuses on interpretation of the theory, while qualitative exploratory research relies on sense making, asking questions and personal interpretation (Bavelas, 1995).

This study uses the confirmatory approach due to the reasons explained above.

4.2 The current study’s paradigm

According to Creswell (2008), researchers need be motivated to acknowledge paradigmatic differences while attentively selecting the methods that provide the greatest opportunity for cross-paradigm communication within the study design. Other authors advise that researchers should adopt a perspective compatible with their research interests and at the same time remain open to the use of other approaches (Hair, Bush & Ortinau 2000). Creswell (2014) suspects that the choice of research method is largely due to factors such as the type of training provided for the researcher, social pressures associated with advisors and colleagues, and preferences for obtaining certain types of insight during the research.

Based on the arguments presented in the previous sections and from the extant literature (critically reviewed and presented in chapters two and three), a number of points can be shown to lead to the choice of an approach for the current study.

First, from the vast body of research on perception level of success (Chow & Cao,2008) it seems that perception level of success research has a dominant theoretical drive which is positivist in nature.

Second, according to Maxwell (2012), interactive design and content analysis define the research approach: the current research purpose is defined as predicting the viability of a model of the perception success behaviour. This purpose requires a more structured, well-defined framework, precise measurement (prior to development of a standardised instrument), establishing relationships between variables, and making inferences from samples to populations, all of which can be seen under the qualitative research umbrella. Furthermore, the content analysis is one of variance analysis, looking for differences between groups; as such, quantitative research is the most appropriate for such an investigation. Additionally, validity issues associated with the type of investigations are to be obtained through statistical conclusion validity, on the construct level (construct validity) and causal validity (control of external variables). In the current study the external variable under investigation is experience with agile methodology.

Third, the major thrust of the current research project is to test hypotheses related to the proposed model extension, as well as a number of hypothesised relationships that were previously established in the perception success context; hence, the theoretical thrust of the current research is deductive in nature and the use of inductive reasoning is excluded from the interpretation of results when compared to previous findings from literature.

Fourth, the current study follows a confirmatory strategy of research, one that envisions empirical analysis as a process of confirming or disproving previously stipulated hypotheses in the perception success context.

Fifth, the current study aims to conduct a number of group comparisons, using the Structure Equation Modelling technique, which is utilised only through statistical packages. Moreover, the researcher is versed in statistics, which makes it a personal preference to work with a quantitative approach. However, in compliance with Hall & Howard (2008) advocates, the researcher is not ignorant of the limitations of one approach and the benefits of mixed methods research. Hence, following the categorisation of Johnson et al., (2007) and although the current research is quantitative, the current research is also mixed. To elaborate on this point, Morgan's (1998) categorisation can be used to establish that the current research is quantitative in principal with a preliminary qualitative study using focus groups to help adjust culturally sensitive versions of perception success of the questionnaires' content.

The remaining sections of this chapter are devoted to defining the research conceptual framework development, research design and methods developed for this study.

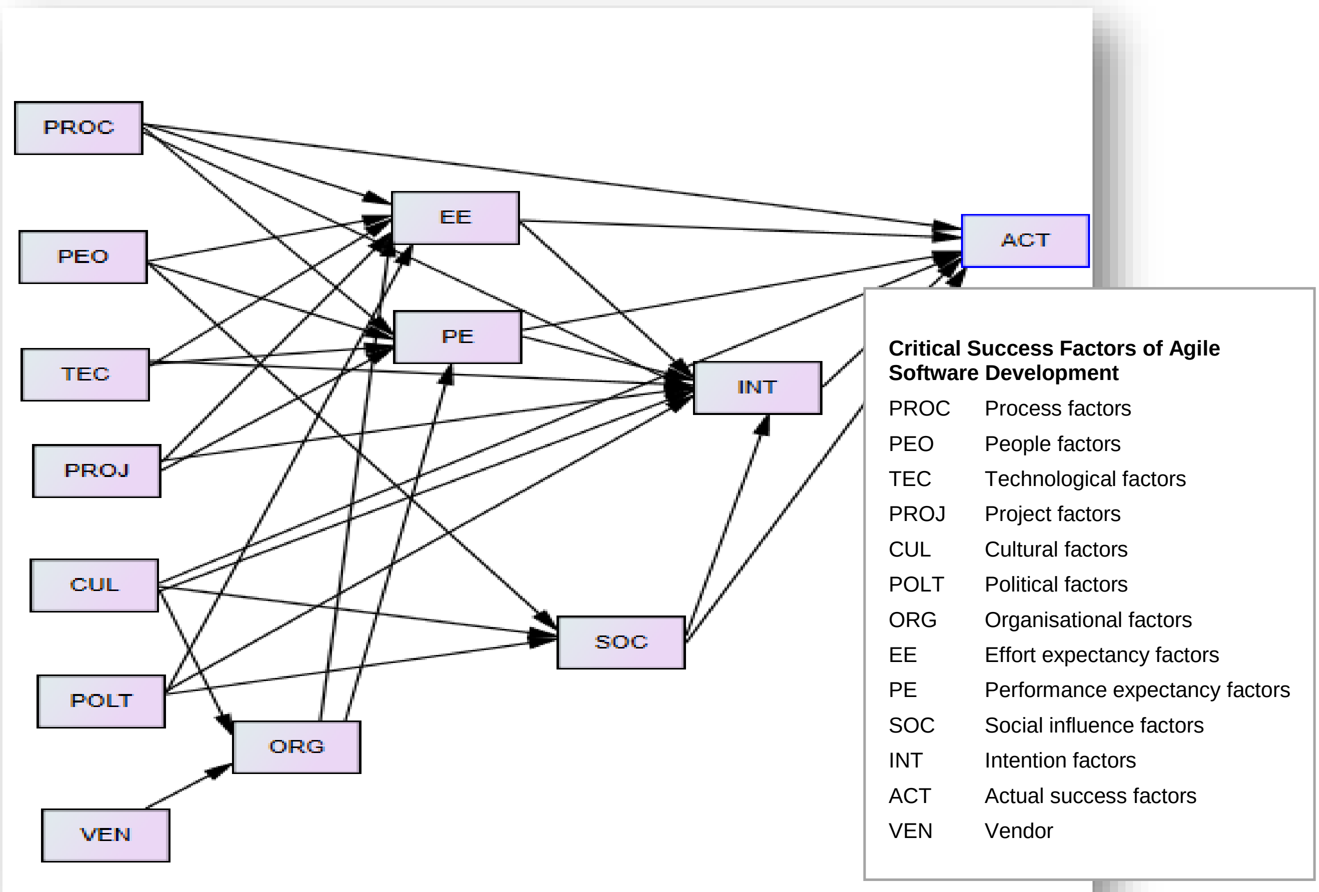


Figure 4.4:The study's proposed research model based on the chapter 2 and chapter 3 literature review

The conceptual framework

The conceptual framework used in this thesis was based on and adapted from TRA, TBP, UTAUT and critical success factors, just to mention a few (Ajzen & Fishbein, 2000; Davis, 1989; Sharma & Kanekar, 2007; Williams, Rana, Dwivedi & Lal, 2011; Yu, 2012; Chow & Chen, 2008; Zhou, Lu & Wang, 2010). The purpose of this study is to evaluate and provide insight into the critical success factors of agile methodologies of software development projects in South Africa using the adapted theories from TRA, TBP, UTAUT and critical success factors, just to mention a few, through the use of a survey completed by agile professionals (agile practitioners) and software managers. A hypothesis research model (adapted from TRA, TBP, UTAUT and critical success factors, just to mention a few) is used to determine the main factors contributing to the actual success of agile methodologies to provide guidelines to agile practitioners, and software managers with regards to software project success (see Figure 4.4).

The relationship between the proposed research model with figure 3.4 and figure 2.12 is that success factors which are discussed in figure 2.12 have direct influence as independent and dependent variables towards the dependent variables in figure 3.2 as supported by Chow and Cao, (2008)..

4.2.1 Research questions

The research questions addressed by the study are:

Primary research questions

1. What are the critical success factors that influence the success of agile software development projects using agile methodologies?

Secondary research questions

1. How do agile professionals perceive the adoption of agile software development projects in South Africa
2. What is the most appropriate theoretical framework which can be adapted to model the critical success factors of agile software development projects?
3. How can critical success factors be structured into a framework that can inform agile professionals and the community?

4.2.2 Research hypotheses and assumption

4.2.2.1 Key decision variables hypothesised causal relationships

a) Social Influence, and intention, and success of agile software development projects

In their study for UTAUT Venkatesh et al. (2003) established that when social influence is moderated by age, gender, experience with technology and voluntariness of use positively influenced behavioral intention to use technology.

This leads to the following hypothesises:

H1: Social influences have a positive effect on intention of agile methodologies for the success of software development projects.

Cabrera, Collins and Salgado, (2006) found that social influence had a significant impact on the success of agile software development projects. If programmers believe that they have the ability to use an agile methodology, they are more likely to use it for software development, and engage in associated knowledge management activities which may not be specified by the job requirements. Hence, programmers with high social influence will be able to take full benefit of the agile methodology to perform their tasks, and also find it easier to use.

H13c: Social Influence has a positive effect on the success of agile software development projects. This leads to the following hypothesises:

H1: Social influences have a positive effect on intention of agile methodologies for the success of software development projects.

H13c: Social Influence has a positive effect on the success of agile software development projects.

b) Performance expectancy, intention, and success of agile software development projects

The factors that affects agile software development projects include perceived usefulness, and perceived ease of use. Perceived usefulness and perceived ease of use have been discovered to affect user perceptions of different technology innovations, such as mobile commerce and agile software development projects (Venkatesh et al, 2003; Chan & Thong, 2009).

In this context, perceived usefulness refers to the degree to which an individual expects that adopting a software development methodology will improve their job performance (Hardgrave et al., 2003). Prior studies on acceptance of software development methodologies (Hardgrave, et al., 2003) have found that perceived usefulness is a significant factor in predicting the intention to use, and success of software development methodologies. Prior research has suggested that the more a methodology is perceived as enabling an increase in job performance, the more likely that it will be accepted.

Venkatesh et al. (2003) discovered that when performance expectancy is moderated by gender and age, intention is positively influenced to effectively use the system. Davis et al. (1992), Davis (1989), and Venkatesh and Davis (2000) support this as they found that perceived ease of use -- which is regarded as effort expectancy in some research – has a positive influence on intention to effectively use the technology. Venkatesh et al (2003) regarded perceived usefulness as performance expectancy.

This leads to the following hypothesises:

H2: Performance expectancy has a positive effect on agile intention for the success of agile software development projects.

H13a: Performance expectancy has a positive effect on the success of agile software development projects.

c) Effort expectancy, and intention, and success of agile software development projects

Perceived ease of use is defined as the degree to which an individual believes that using a particular agile methodology would be free of physical and mental effort (Moore and Benbasat, 1991). Venkatesh et al. (2003) regarded perceived ease of use as effort expectancy. Effort expectancy has been found to be an important factor influencing the intention to adopt an innovation in previous studies on several technologies (Davis, 1989). The principles of agile methodologies are different from those of traditional software development methodologies in most cases. Thus, effort expectancy may have a greater influence when ICT professionals switch from traditional methodologies to agile methodologies than when ICT professionals switch to another traditional software development methodology (Chan & Thong, 2009). ICT professionals are more likely to accept agile methodologies when the methodologies are easy to use.

Based on the argument formulated by Venkatesh and Davis (2000), Davis et al. (1992) and Davis (1989), effort expectancy positively impacts intention when moderated by gender, age and experience with technology.

This leads to the following hypothesises:

H3: Effort expectancy has a positive effect on agile intention for the success of agile software development projects.

H13b: Effort expectancy has a positive effect on the success of agile software development projects.

d) Culture Factors

According to Fishbein and Ajzen (2005), the culture of users affects their social influences and normative beliefs. This includes subjective influences such as the users' attitudes, beliefs, values, behaviours and opinions about the objectives that they use daily (Koch, 2005; Nerur et al., 2005). This suggests that culture, indirectly or directly, orders ways of living, and governs peoples' communications and their social lives. Therefore, culture regulates the pyramid for decision-making and sets the values for users' collaboration thus positively influencing their social influence.

This leads to the hypothesis below:

H9a: Culture has a positive effect on social influences for the success of agile software development projects.

Culture is formed as users become familiar with their surroundings. Familiarising themselves with their settings, they are able to address common social challenges and adopt the essentials that lead to success (Boehm & Turner, 2005; Triandis, 1980a). Consequently, the organisational culture and the particular social community in any organisation vitally impact on users' attitudes and behaviours towards learning and regulating new inventions in the organisation (Koch, 2005; Triandis, 1980). Therefore, users within various cultural settings will have diverse behavioural intentions, meaning attitudes towards the success of agile.

This leads to the following hypothesis:

H9b: Culture has a positive effect on agile intention for the success of agile software development projects.

e) Culture and Political Factors

Culture and politics are external factors that have previously been argued to have a positive direct impact on the success of the agile software development projects (Koch, 2005; Venkatesh et al., 2003). For example, if agile professionals continuously support junior software developers and train them, the organisation will in turn assist other junior software developers to attain self-efficacy with the scheme. Hence, the external factors impact on the organisation with regards to culture and political aspects to assist junior software developers to have improved motivation regarding agile's benefits and performance, so that they appreciate it and will eventually effectively use it for the success of agile software projects (Lalsing et al., 2012).

Several organisations, particularly software development companies, have their top management politically selected (Koch, 2005; Wan & Wang, 2010). Such political appointees have an influence over software developer staffing, particularly in relation to those in executive positions. Hence, politics might lead to an interruption of the entire integration efforts of software projects. Therefore, it is important to include politics as a critical success factor of the usage of agile methodologies in that choosing the right management and agile professionals is crucial to success for agile software development projects.

This leads to the following hypotheses:

H6a: Culture has a positive effect on the success of agile software development projects.

H6b: Political factors have a positive effect on the intention of agile software development projects.

H6c: Political factors have a positive effect on the organisational success factors of agile software development projects.

Furthermore, it is presumed that political impacts are entrenched in the users' day to day events. That means that social events are not only affected by the people they perceive to be significant in their specific job activities, but moreover by their environments (Venkatesh et al., 2003). Hence, basing on this understanding hypothesis H7a and 7b were drawn.

Therefore, hypotheses H7a and H7b were formulated.

H7a: Political factors have a positive effect on agile professionals' social influence for the success of agile software development projects.

H7b: Political factors have a positive effect on agile professionals' effort expectancy for the success of agile software development projects.

f) Intention Factors

The dependent variable in this study is agile professionals' intention to use agile methodologies (Hardgrave et al., 2003). Intention is normally used as an indicator of the use and acceptance of technology adoption in UTAUT (Venkatesh et al., 2003). Intention has become the de facto measure for evaluating the acceptance of an invention and has continually proven to be a strong predictor of actual use (in other words, the success of software projects) (Venkatesh et al., 2003). According to Ajzen (1991), intentions are expected to capture the motivational factors that have influence on a behaviour; they are signs of how hard people are willing to try, of how much of an effort they are planning to exert in order to perform the behaviour, based on past experience. These intentions remain behavioural dispositions until, at the suitable time and opportunity, an effort is made to decode the intention into action (Venkatesh et al., 2003).

Curtis and Payne (2008) and Tibenderana et al. (2010) found UTAUT very predictive for intention which has a positive impact on actual behaviour.

This leads to the following hypothesis:

H4: Intention have a positive effect on the success factors of agile software development projects.

g) Organisation Factors

Numerous researchers, such as DeLone and McLean (2002) and Curtis and Payne (2008), suggest that organisational factors play an important role in the use of agile methodologies which ultimately leads to the success of agile software development projects. Similarly, research on agile methodologies showed that some of the top critical success factors of agile methodologies are organisational factors (Koch, 2005; DeLone & McLean, 2002). Therefore, scientists have found that top executive or management support is one of the first priorities with regards to agile methodologies (Chow & Cao, 2008). These scientists claim that executives or top management need to show tangible support for the implementation of agile methodologies in the project (Ramesh, Cao & Baskerville, 2010). This can be in the form of giving help or support to users (such as team members of the group), hiring the services of consultants, training users, and providing psychological, financial and moral support.

Several researchers, such as Curtis and Payne (2008) and Venkatesh et al. (2003), called this kind of help or support “facilitating conditions”. Venkatesh et al. (2003) empirically recognised that facilitating conditions have a direct effect on the success of agile software projects. Previous studies conducted by Davis (1989), Rogers (2008) and Venkatesh and Davis (2000) have shown that help or support for users is directly proportional to self-efficacy which led to an increase in the users’ capability. Also Compeau and Higgins (1995b) noted that when users perceive that they have enough competence to use the technology, their perceived ease of use increases. This has a positive influence on their intention to use agile methodologies.

This leads to the following hypotheses:

H10a: Organisational factors have a positive effect on effort expectancy for the success of agile software development projects.

H10b: Organisational factors have a positive effect on performance expectancy for the success of agile software development projects.

H10c: Organisational factors have a positive effect on the success of agile software development projects.

h) People Factors

According to Triandis (1980b), people's beliefs play a significant part in whether users believe a system is easy or difficult to use. This explains why, when people believe that a system is easy to implement, they will implement it, and if not why they will avoid it (DeLone & McLean, 2002). So when users perceive a system as easy to use and they go ahead and use it. This means that it is also easier for them to understand its benefits. Likewise, users' viewpoints have an effect on their societal beliefs. Bossini and Fernández (2013), and Boehm and Turner (2003c), have noted that, individual or people factors are critical success factors affecting software projects using agile methodologies. These comprise trust, conflict, commitment, cooperation, communication, dependency and satisfaction among individual group members, managers, whole groups, sponsors and customers.

This leads to the following hypotheses:

H8a: People factors have a positive effect on effort expectancy factors for the success of agile software development projects.

H8b: People factors have a positive influence on performance expectancy factors for the of agile software development projects.

H8c: People factors have a positive influence on social influence factors for the success of agile software development projects.

H8d: People factors have a positive effect on the overall perceived success of agile software development projects.

i) Technological, Process and Project Factors

Technology Factors

This research shows that technological factors can have direct influence on intention and on the success of the agile software development project respectively (Marnewick & Labuschagne, 2009; Augustine, Misra et al., 2009; Ceschi, Sillitti, Succi & De Panfilis, 2005, Venkatesh et al., 2003). UTAUT decision variables (performance expectancy and effort expectancy) impact on the perception of the agile practitioner's intention regarding the success of agile software development projects (Venkatesh et al., 2003). This suggests that both performance expectancy and effort expectancy are associated with the technological aspect of the agile software development project, as well as on how agile practitioners perceive the project. Davis (1989), Davis et al. (1992), Venkatesh and Davis (2000) and Venkatesh et al. (2003) all recognised that both performance expectancy and effort positively impact intention (see Figure 4.2).

This leads to the following hypotheses:

H5a: Technological factors have a positive effect on intention for the success of agile software development projects.

H5b: Technological factors have a positive effect on effort expectancy for the success of agile software development projects.

H5c: Technological factors have a positive effect on performance expectations for the success of agile software development projects.

H5d: Technological factors have a positive effect on the success of the agile software development projects.

Process Factors

This research shows that process factors can have direct influence on intention and on the success of the agile software development project respectively (Marnewick & Labuschagne, 2009; Augustine, Misra et al., 2009; Ceschi, Sillitti, Succi & De Panfilis, 2005, Venkatesh et al., 2003). UTAUT decision variables (performance expectancy and effort expectancy) impact on the perception of the agile practitioner's intention regarding the success of agile software development projects (Venkatesh et al., 2003). This suggests that both performance expectancy and effort expectancy are associated with the process aspect of the agile software development project, as well as on how agile practitioners perceive the project. Davis (1989), Davis et al.

(1992), Venkatesh and Davis (2000) and Venkatesh et al. (2003) all recognised that both performance expectancy and effort positively impact intention (see Figure 4.2).

H11a: Process factors have a positive effect on effort expectancy for the success of agile software development projects.

H11b: Process factors have a positive effect on performance expectations for the success of agile software development projects.

H11c: Process factors have a positive effect on intention for the success of agile software development projects.

H11d: Process factors have a positive effect on the success of agile software development projects.

Project Factors

This research shows that, project factors can have direct influence on intention and on the success of the agile software development project respectively (Marnewick & Labuschagne, 2009; Augustine, Misra et al., 2009; Ceschi, Sillitti, Succi & De Panfilis, 2005, Venkatesh et al., 2003). UTAUT decision variables (performance expectancy and effort expectancy) impact on the perception of the agile practitioner's intention regarding the success of agile software development projects (Venkatesh et al., 2003). This suggests that both performance expectancy and effort expectancy are associated with the process aspect of the agile software development project, as well as on how agile practitioners perceive the project. Davis (1989), Davis et al. (1992), Venkatesh and Davis (2000) and Venkatesh et al. (2003) all recognised that both performance expectancy and effort positively impact intention (see Figure 4.2).

H12a: Project factors have a positive effect on performance expectations for the success of agile software development projects.

H12b: Project factors have a positive effect on effort expectancy for the success of agile software development projects.

H12c: Project factors have a positive effect on intention for the success of agile software development projects.

H12d: Project factors have a positive effect on the success of agile software development projects.

4.2.3 The effect of moderating factors

Venkatesh et al. (2003) describe the four moderating factors that form part of UTAUT. These have a moderating influence of the model's decision variables and are age, gender, experience with technology (such as agile methodologies) and voluntariness of use.

4.2.3.1 Gender

Gender was established to have a moderating influence on social influence, effort expectancy, and performance expectations (Venkatesh et al., 2003). Gender was retained in this research under the assumption that it will positively impact on the three constructs as was in the case of the inventive UTAUT model.

This leads to the following hypothesis:

H14: Gender moderates the relationships among the proposed model constructs

H14a: Gender positively moderates the relationships between individual performance expectancy factors and overall success of agile software development projects.

H14b: Gender positively moderates the relationships between individual effort expectancy factors and overall success of agile software development projects.

H14c: Gender positively moderates the relationships between project factors and performance expectancy factors for the success of agile software development projects.

H14d: Gender positively moderates the relationships between people factors and performance expectancy factors for the success of agile software development projects.

4.2.3.2 Age

According to Venkatesh et al. (2003), younger users will use technology more often and contentedly than their aged counterparts. This suggests that young people tend to explore new innovations mostly because of their curiosity or social influences, while older people tend to rely on their experience (Boehm, 2002; Dyba & Dingsøyr, 2008). Age is considered a moderating factor in this research, and its influence on effort expectance, performance expectancy and social influence need to be evaluated.

This leads to the following hypotheses:

H15: Age moderates the relationships among the proposed model constructs

H15a: Age positively moderates the relationships between process factors and performance expectancy factors for the success of agile software development projects.

H15b: Age positively moderates the relationships between performance expectancy factors and intention factors for the success of agile software development projects.

H15c: Age positively moderates the relationships between effort expectancy factors and intention factors for the success of agile software development projects.

4.2.3.3 Experience

According to Venkatesh et al. (2003), Boehm (2002), and Dyba and Dingsøyr (2009), experience with the technology in use (in this case, agile tools) is important. They found that the more experience a user has with the technology, the more likely that user will use the technology effectively. Therefore, this study hypothesises those users that have experience with agile will effectively use it, as compared to their counterparts with no such prior experience (Bahli, Benslimanne, & Yang, 2011; Bishop & Deokar, 2014; Dyba & Dingsoyr, 2009; Wolff, 2012).

This leads to following hypotheses:

H16: Experience moderates the relationships among the proposed model constructs

H16a: Experience positively moderated the relationships between effort expectancy factors and success factors of the agile software development projects.

H16b: Experience positively moderated the relationships between process factors and performance expectancy factors of the agile software development projects.

H16c: Experience positively moderated the relationships between process factors and effort expectancy factors of the agile software development projects.

4.2.3.4 Voluntariness of use

Voluntariness of use is not considered particularly important in regard to this study. The research study focuses on agile professionals who are using agile methodologies specifically. In place of voluntariness of use, an alternative moderating factor is presumed to be relevant to the current research, namely the education level of the employees within an organisation (Dyba & Dingsøyr, 2008).

4.2.3.5 Educational level

With regards to level of education, numerous investigators, such as Asnawi, Gravell & Wills (2012), Hazzan and Dubinsky (2010), and Dyba and Dingsøyr (2008), have highlighted that when organisations train agile practitioners, there is a high possibility of the agile software development project succeeding. This suggests that there is a positive correlation between the level of education and the success of the agile software development project. Asnawi et al. (2012) and Rodríguez, Markkula, Oivo and Turula (2012) emphasise the need and significance of support, such as training, which top management can offer to the team members. This leads to the point that, when agile practitioners have sufficient skill and the practical knowledge of using the technology, both their performance expectancy and effort will increase or decrease.

This leads to the following hypotheses:

H17: Educational level moderates the relationships among the proposed model constructs.

H17a: Educational level positively moderates the relationships between effort expectancy factors and success of agile software development projects.

H17b: Educational level positively moderates the relationships between process factors and performance expectancy factors for the success of agile software development projects.

H17c: Educational level positively moderates the relationships between process factors and effort expectancy factors for the success of agile software development projects.

4.2.4 Actual and continued use

The model in Figure 4.2 investigates the actual use of agile methodologies, however intention was also included to investigate continued usage which is necessary for the success of agile software development projects. From the model it is assumed that technical factors, political factors and cultural factors will influence intention to use agile methodologies. This implies that agile professionals may initially accept and adopt agile methodologies but that their attitude to use it may change with continued usage. In this study, success of agile software development projects is regarded as actual behaviour which can be achieved through use of agile methodologies.

4.3 Research Instrument development and design

This section explains the decision variables measurement for the suggested research model, the instrument translation into abbreviations, instrument development, and instrument pretesting.

4.3.1 Instrument development

The research instrument consisted of three parts and a cover letter which indicated the purpose of the study and details for the research. The first part covered the interview questions, the second part focused on the demographic variables such as age, gender, level of experience and level of education and the third part included questions dependant on the respondents' answers regarding the constructs which were discovered in the literature review such variables including performance expectancy, effort expectancy, and organisational, project, process, technological, people, intentional and actual factors as shown in Appendix A. Part three used a five point Likert scale, (where 1 represents strongly disagree, 2 represents disagree, 3 represents neutral, 4 represents agree and 5 represents strongly agree) for the questions as shown Appendix A and Table 4.5 below are adapted from Moore and Benbasat, (1991). Both open-ended and closed-ended questions were used in this research. Open-ended questions were used to construct the closed-ended questions. Construct definition is shown in table 4.6.

Table 4.5: The three sections of the questionnaire

Stage	Section	Decision variables
1	A:	Biographical information
2	B	Organisational factors
		Project factors
		People factors
		Intentional behaviour factors
		Technological factors
		Actual success factors
		Effort expectancy factors
		Performance expectancy factors
		Process factors
		Political factors
		Social factors
		Cultural factors
3	C	Open-ended questions

Table 4.6: Construct definition

Construct Description Source	Construct Description
Process factors	Process factors refer to a series of actions that you take in order to achieve a successful agile software development project (Chow & Cao, 2008)
People factors	People factors refers to how management empower people by supporting realistic goals, ownership, flexibility and shorter feedback cycles (Cockburn & Highsmith, 2001).
Political factors	Political factors refers to agile professional’s social activities that are not only influenced by the people they perceive important.
Cultural factors	Culture factors denotes the complex system of meanings, symbols, and assumptions about what is legitimate or illegitimate, good or bad, that inspires the prevailing practices and norms in an agile society (Huisman, Nd; Iivari & Iivari, 2011; Rogers, 1983; Strode, Huff & Tretiakov, 2009).
Project factors	People factors refers to how management empower people by supporting realistic goals, ownership, flexibility and shorter feedback cycles (Cockburn & Highsmith, 2001).
Technical factors	Technical factors refers to the operation of software projects, or techniques that may be applied when developing software projects that will lead to successful technical performance (Rogers, 1983; Sutharshan, 2011).
Organisational factors	Organisational factors refer to those factors related to an organisation that may affect the success of agile methodologies (Chow & Cao, 2008; Nasehi, 2013).

Vendor factors	Vendor factors refers to the role of external shareholders and the environment that surrounds the users of the agile methodologies (Chow & Cao, 2008; Nasehi, 2013).
Effort expectancy factors	Effort expectancy factors refers to the degree to which individual team members perceive agile methodologies as easy to use Riemenschneider, 2003; Venkatesh et al., 2003).
Performance expectancy	Performance expectancy factors describes the degree to which an individual team member believes that using agile methodologies is critical to them accomplishing their work processes (Hardgrave, Davis & Riemenschneider, 2003; Venkatesh et al., 2003).
Social influence factors	Occurs when one's emotions, opinions or behaviour is affected by others (Riemenschneider, 2003; Venkatesh et al., 2003).
Intention factors	Intention factors refers to something that agile professionals want and plan to do for the project to be successful (Chow & Cao, 2008, Venkatesh et al., 2003).
Actual success factors	Actual success factors refers to achievement of practicing agile methodologies within a specified period of time or within a specified parameter (Chow & Cao, 2008, Venkatesh et al., 2003).

4.3.2 Constructs' measurement

The terms constructs and decision variables are regarded as having the same meaning.

4.3.3 Instrument translation

The research instrument (questionnaire) was translated from English to abbreviation by the researcher see Table 4.7. These are applicable to the questionnaire in Appendices A and B and discussions in Chapters 4, 5, 6, 7 and 8.

Table 4.7: Instrument translation

Abbreviation **Questions**	**English Questions**
Process_1	Question 1: Process factors
Process_2	Question 2: Process factor s
Process_3	Question 3: Process factor s
Process_4	Question 4: Process factors
Process_5	Question 5: Process factors
Process_6	Question 6: Process factors
Process_7	Question 7: Process factors
Org_1	Question 1: organisational factors
Org_2	Question 2: organisational factors
Org_3	Question 3: organisational factors
Org_4	Question 4: organisational factors

Abbreviation Questions	**English Questions**
Project_1	Question 1: Process factors
Project_2	Question 2: Process factors
Project_3	Question 3: Process factors
Project_4	Question 4: Process factors
People_1	Question 1: People factors
People_2	Question 2: People factors
People_3	Question 3: People factors
People_4	Question 4: People factors
Tech_1	Question 1: Technological l factors
Tech_2	Question 2: Technological factors
Tech_3	Question 3: Technological factors

Abbreviation **Questions**	**English Questions**
Orgl_5	Question 5: organisational factors
Org_7	Question 7: Organisational factors
Org_8	Question 8: Organisational factors
Performance_1	Question 1: Performance factors
Performance_2	Question 2: Performance factors
Performance_3	Question 3: Performance factors
Performance_4	Question 4: Performance factors
Actual_1	Question 1: Actual success factors
Actual_2	Question 2: Actual success factors
Actual_3	Question 3: Actual success factors
Actual_4	Question 4: Actual success factors
Actual_5	Question 5: Actual success factors
Actual_6	Question 6: Actual success factors
Actual_7	Question 7: Actual success factors
Culture_1	Question 1: Culture factors
Culture_2	Question 2: Culture factors
Culture_3	Question 3: Culture factors
Culture_4	Question 4: Culture factors
Culture_5	Question 5: Culture factors
Social_1	Question 1: Social factors
Social_2	Question 2: Social factors
Social_3	Question 3: Social factors
Social_4	Question 4: Social factors
Social_5	Question 5: Social factors
Social_6	Question 6: Social factors
Social_7	Question 7: Social factors
Social_8	Question 8: Social factors

Abbreviation Questions	**English Questions**
Tech_4	Question 4: Technological factors
Tech_5	Question 5: Technological factors
Tech_6	Question 6: Technological factors
Tech_7	Question 7: Technological factors
Intention_1	Question 1: Intention success factors
Intention_2	Question 2: Intention success factors
Intention_3	Question 3: Intention success factors
Intention_4	Question 4: Intention success factors
Intention_5	Question 5: Intention success factors
Intention_6	Question 6: Intention success factors
Intention_7	Question 7: Intention success factors
Effort_1	Question 1: Effort expectancy factors
Effort_2	Question 2: Effort expectancy factors
Effort_3	Question 3: Effort expectancy factors
Effort_4	Question 4: Effort expectancy factors
Effort_5	Question 5: Effort expectancy factors
Effort_6	Question 6: Effort expectancy factors
Effort_7	Question 7: Effort expectancy factors
Political_1	Question 1: Political factors
Political_2	Question 2: Political factors
Political_3	Question 3: Political factors
Political_4	Question 4: Political factors

4.4 Research design

Creswell (2014) notes that research design provides a model for the collection and analysis of data. Research designs include consideration of research approaches, as well as the practices of collection of data, and can include precise instruments such as structured interviews or self-administered questionnaires (De Vaus & de Vaus, 2001). Creswell (2007) identified that the purpose of a research design is to ensure that the evidence acquired permits one to answer the initial question as explicitly as possible. De Vaus and de Vaus (2001) stated that research design includes a series of rational decision-making selections concerning the purpose of the study (which, in this study, was regarded as exploratory and hypothesis evaluating). Furthermore, the research locality, the type of examination, the degree of researcher intervention, time perspective, and the level to which the data will be analysed (such as unit of analysis) should be included as well as the purpose of the research design. Additionally, choices have to be made on the sampling design, how variables will be measured, how data is to be collected, and how data is analysed to evaluate the hypotheses.

Following De Vas's (2001) definition of research design, this study is carried out for the purpose of evaluating the hypotheses derived from the conceptual model of the research study. The research study used hypotheses evaluation to explain the nature of specific causal relationships and to discover the differences among groups. Hypotheses evaluating provides an improved understanding of the causal relationships that occur between the variables.

The time characteristic is a significant part of the design and implementation of a research study (Creswell, 2014). Research is typically either a cross–sectional or a longitudinal study (Creswell, 2014). A longitudinal study is based on the unit of analysis which is observed over a lengthy period of time (Creswell, 2007). Additionally, longitudinal studies permit changes to be presented over a period of time. It is challenging to undertake longitudinal research due to the large scale of the research (Creswell, 2014). Longitudinal research study is also often expensive in terms of both cost and time. Hence, longitudinal research was not used in this study. Instead a cross-sectional research approach was adopted. A cross-sectional research study is where the unit of analysis is observed at one point in time (Creswell, 2014). De Vaus and de Vaus (2001) note that cross-sectional research is carried out once only and represents a snapshot in time. Following the snapshot, the causal relationships between variables are investigated. In this research, a cross-sectional study was considered the most suitable as the study involved a fairly large-scale survey administered in South Africa which was carried out once only.

4.5 Research Approach

Research approach explains the pattern of ideas, techniques and assumptions that distinguish quantitative and qualitative research (Creswell, 2014). Both qualitative and quantitative research were used in this study. The study focused mainly on a quantitative approach and to a less extent on a qualitative approach. The quantitative approach is summarised as an inquiry into a social or people challenges, grounded on evaluating a research framework comprised of variables, measured with figures and analysed with statistical techniques methods, in order to determine whether the extrapolative generalisations of the framework embrace the truth (Oates, 2006). According to Oates (2006), the major phases in a quantitative research process can be abridged as shown in Figure 4.5.

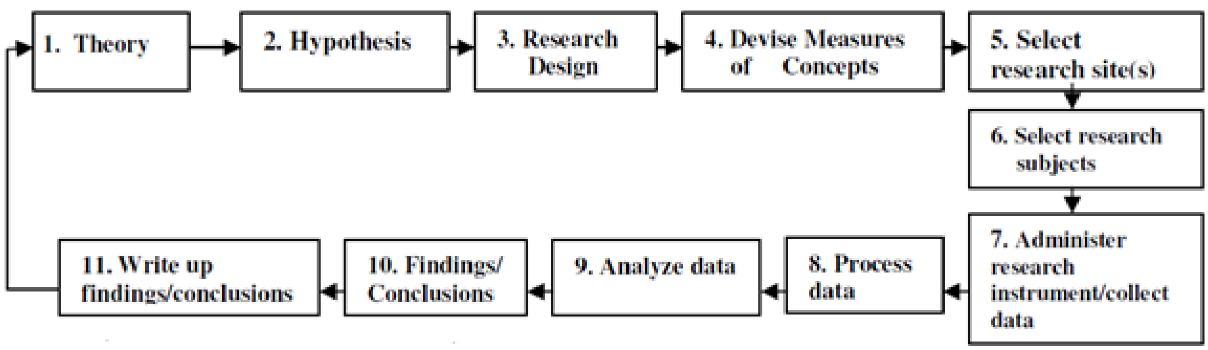


Figure 4.5: The process of quantitative research (adapted from Oates, 2006)

Figure 4.6 below illustrates how the quantitative and qualitative aspects of the study work together. The quantitative part consists of the following steps: motivation, literature review, research questions, survey, and questionnaires. The qualitative research process consists of motivation, literature review, research questions, survey and interview.

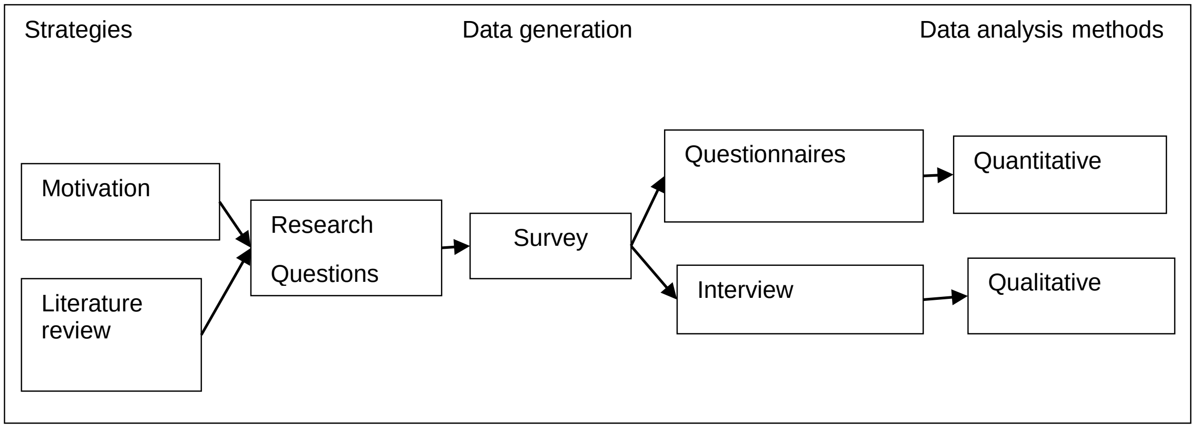


Figure 4.6: The research process (adapted from Oates, 2006)

4.6 Research techniques

Below is a description of the research methodology outlining the target population, sampling, data collection, and data analysis (Leedy & Ormrod, 2005).

4.6.1 Target population

A target population is the whole group under study as stated by the research (Creswell, 2013).

The target population of this research included Information Technology (IT), programmers, software development and IT project managers who are regarded as agile professionals and are based in South African organisations that have been involved with agile methodologies.

4.6.2 Sampling

Sampling is the process of choosing units such as organisations or a group of people from a particular population of interest so that, by studying the sample, we can generalise the results to the target population from which they were selected (Creswell, 2013). According to Creswell (2013), a sample includes groups selected with the purpose of finding out something about the complete population from which they are engaged. The goal of scientific research is to explain and describe the nature of a population, a group or class of subjects, variables, concepts, and phenomena. In most circumstances, the entire population cannot be involved due to time and resource constraints (Creswell, 2013). The usual phases to be followed in these instances are thus to take a representative sample from the population (in this research from IT, programmers, software developers and IT experts).

4.6.2.1 Sampling methods

There are several diverse methods of sampling and these can be grouped into probability and non-probability sampling techniques (Creswell & Clark, 2007). Probability sampling is a technique whereby all affiliates of the target population have a non-zero probability of being selected (Fink, 2015). Probability sampling has the benefit that it is possible to determine the sampling error, which is the degree to which a target population might vary from the entire population (Kolb, 2008). In contrast, non-probability sampling is a technique in which units of the sample are selected based on personal or convenience judgment (Creswell, 2013). In this research study non-probability sampling was used due to the fact that some units in the population were more likely to be selected than others (IT, programmers, software developers and IT experts rather than from non-IT and non-programmers).

Table 4.8: Techniques of non-probability sampling (Babbie, 1990)

Techniques	Description
Convenience sampling	Also called haphazard or opportunity sampling, this is a technique in which a sample is drawn from that part of the population that is close to hand, readily available, or involves people, animals or units that are most conveniently available.
Purposive sampling	Also referred to as judgmental sampling. An experienced individual selects the sample based on his or her judgment about some appropriate characteristics required of the sample members.
Quota sampling	Is whereby the population is divided into mutually exclusive subdivisions such as in stratified sampling, and then a non-random set of observations is selected from each subdivision to meet a predefined quota.
Snowball sampling	In snowball sampling, you start by finding a few participants that match the standards for inclusion in your study, and then ask them to recommend others they know who also meet your selection standards.

This research uses purposive and convenience sampling.

Purposive sampling is a sampling technique in which researcher relies on his or her own judgment when choosing members of population to participate in the study. Purposive sampling is a non-probability sampling method and it occurs when elements selected for the sample are chosen by the judgment of the researcher. Researchers often believe that they can obtain a representative sample by using sound judgment, which will result in saving time and money.

The sample in this research is considered purposive for the following reasons:

- The respondents represent information-rich cases for in-depth study of the qualitative and quantitative components of the research (as shown in Table 4.5).
- The sample quantity was deemed representative which implied well-informed and informative responses on the phenomena being examined (Creswell, 2013).

Convenience sampling is described as a type of non-probability sampling method which includes the sample being drawn from that part of the population which is close at hand (Creswell, 2013). It is thus a sample population that is selected because it is readily available and convenient.

The sample in this study is considered convenience sampling because:

1. This sample was selected due to the accessibility and proximity of the participants.
2. The most well-known organisations were selected.
3. The respondents of the pilot study cope well with the requirements of the questionnaire.

The reason for choosing South Africa was that it was easily accessible and contains a suitable population sample, which was required for the study.

4.6.2.2 Sample size

The sample size is important to consider in any research study. According to Creswell (2013), the size of a sample is based on the type of research being done, although practical limitations may have an effect. Therefore, by specifying identified inclusion criteria, the sample becomes homogeneous, which means that there is not much variance within the sample, allowing a smaller sample size (Creswell & Clark, 2007). According to Acharya, Prakash, Saxena & Nigam (2013), the most suitable sample size relies on the following aspects: nature and size of the population under study, resources, budget, time available, the required accuracy of the study, and the significance of the results. In this research, the sample constitutes 460 respondents. The study followed the guidelines and standards set out Barlett, Kotrlik & Higgins (2001) for providing a suitable sample size.

4.6.3 Data collection methods

4.6.3.1 Participants

When performing research it is important to think carefully about how one could select the respondents. The respondents must have the ability to offer the information needed. Similarly, the research requires the respondents to have certain simple cognitive skills (Babbie, 1990), such as understanding of non-verbal and verbal material, certain symbols and conversation rules. In addition, choosing the participants for a study requires the following techniques: deciding how many respondents to include in the study, developing a user questionnaire as shown in appendices A and B, choosing subgroup(s) for the study, and quantifying and defining the characteristics for each subgroup (Creswell, 2013).

In this research, participants were IT experts who were deemed to be able to provide insight into the critical success factors of agile methodologies. Software developers or programmers who use agile methodologies were selected because of their experience. It was felt that software developers or programmers with at least one year experience in agile should be able to provide the required information for the study. Software developers or programmers were selected from different provinces in South Africa. Consequently, different races and cultures were included in the study.

Primary data were used in the research. According to Creswell and Clark (2007) primary data is data collected and linked together for a particular research study to provide meaningful information. Primary data can be collected through interviews, observations, and self-administered instruments (De Villiers, 2012). Primary data were collected through the use of emailed questionnaires and interviews to the target population.

The study consisted of a pilot study and a main study.

4.6.3.2 Pilot study

According to Creswell (2013), a pilot study is able to help in removing unnecessary questions which might arise during the main study. In addition, a pilot study checks the suitability of the research methods and research design, and enables the researcher to make changes, if required, to the questionnaire, for instance by eliminating problems highlighted in the pilot study (Acharya et al., 2013; Chaleunvong, 2009).

A pilot study was carried out with 30 IT experts in Pretoria. The purpose of the pilot study was to collect data to reduce risk and uncertainty. Based on the pilot study, no questions were redesigned.

4.6.3.3 Main study

In the main study 460 questionnaires were completed and 500 questionnaires were submitted for the survey. The completion of the questionnaire included three successive parts. In Parts 1 the respondents were asked to complete the demographic questions aimed at obtaining information about the moderating effects in the proposed model. In Parts 2, several questions on agile methodologies in software project were asked, as shown in Appendix A. In Parts 3, open-ended questions were asked to permit the in-depth study of critical success factors of agile methodologies in software development, as illustrated in Appendix A.

The instruments used to collect data were the following:

a) Questionnaire

The study focused on two groups of people, namely software developers and software development managers. The questionnaire included semi-structured and structured questions. The semi-structured questions enabled the respondents to answer the questions by giving their own opinion as shown in Appendix B (Creswell, 2013). Semi-structured questions enable the respondents to freely and fully express their views with regard to the critical success factors of agile methodologies in software development (Kolb, 2008). The questionnaire was emailed to the respondents from 1 of January 2016 to 29 March 2016. The researcher demonstrated how the questionnaire should be fully completed. This was carried out in order to simplify what was required in the questionnaire through instructions from the researcher. There were 15 software development managers who answered the questions.

A 5-point Likert scale was used to reflect the level of actual success of the respondent. The scale ranges from 1 which represent strongly disagree and 5 which represented strongly agree, with 3 as the neutral point.

b) Interviews

The objective of the interviews was to confirm that the factors discovered in the literature review were the same as those emerging from the South African context. The interviews were carried out with IT management, IT experts and software engineering management. The researcher used factors which were supported both in the literature review and interviews. In total, 15 interviews were completed with IT professionals and management and the themes emerging are shown in Appendix B. Opened ended questionnaire was used for interviews. Interviews were done at the workplace of the interviewees on different dates between 1 March and 31 April 2016.

4.6.4 Data analysis

The goal of the data analysis was to structure the findings as an interpretation of the most critical success factors for South Africa with regards to using agile methodologies in software development. By means of the data analysis, conclusions were drawn, and challenges and successes identified. According to Creswell (2013), data analysis is the most complex of all the phases of a quantitative research study, and yet it receives the least attention in the literature. In order to arrive at findings that transform raw data into new knowledge, it is necessary to engage in proactive and in-depth analytic processing throughout all the stages of the study (De Villiers, 2012). These processes are important in quantitative research and for understanding, interpreting and reading results when utilised in the quantitative research study.

4.6.4.1 Analysis methods

Descriptive statistics were applied to analyse the data and determine the significance of the model using SPSS software (Byrne, Stewart, Kennard, & Lee, 2007). The correlation between the decision variables of the participants with regard to the proposed model was tested to determine whether there was a significant association between decision variables in South Africa. A moderating effect test on experience, education level, gender and age was carried out to test the differences between respondents with regards to the decision variables. Structural equation modelling was used to determine the dominant critical success factors in the proposed model (Pallant, 2013).

The data were analysed for validity based on convergent and content validity using a scale factor analysis (French & Finch, 2006). The convergent validity test measures items for convergence or divergence on the decision variables with factor loading having a fixed value (Golafshani, 2003). The reliability of measurement items was tested using Cronbach’s alpha test for decision variables (Hu & Bentler, 1999). Cronbach’s alpha test enabled the researcher to assess the internal consistency of the proposed decision variables (Pallant, 2013). A structural equation modelling (SEM) test was used to model latent variables under conditions of non-normality with small to medium-sized samples (Pallant, 2013). This method is considered suitable for this type of study since it answers the questions that include multiple regression analysis of factors among a single measured dependent variable and a group of measured independent variable. The SEM method is suitable for comparison of two groups which makes it an appropriate tool for evaluating the hypotheses and achieving the objectives of the research study. Confirmatory factor analysis is a multi-group analysis that assesses equivalence, especially if the individual group represent samples and enables assessment of metric invariance and scalar invariance (Barrett, 2007; Byrne, 2013). Also invariance assists in evaluating equivalence among the samples in this research study (Cheung & Rensvold, 2002). In addition, latent variables are used when differences are evaluated among groups (Iacobucci, 2010).

4.7 Ethical considerations

According to Creswell (2014), researchers must safeguard their research respondents, promote the integrity of research, develop confidence with participants, guard against misconduct and impropriety that might reflect on their organisations, and cope with new problems. It is the duty of the researcher to ensure that ethical standards are adhered to.

For this study, ethical clearance was obtained from the University of South Africa to ensure that the study maintained research integrity embodying a range of good research practices and conduct, including accuracy, intellectual honesty, fairness and protection of human participants involved in piloting the research (Leedy & Ormrod, 2005).

4.7.1 Permission to conduct the research

The researcher obtained permission to conduct the study from the relevant authorities of the industrial companies. The request for permission to conduct the study was forwarded to the University of South Africa (see Appendix D), and permission was granted (see Appendix C).

4.7.2 Informed consent forms

The ethical requirement of informed consent means that participation in the research study is voluntary (Creswell, 2013). Informed consent forms were thus sent to IT companies and development companies who contributed to the research to sign, affirming that they decided to participate of their own free will (see Appendix D). It was important for respondents to understand that the study would be investigating the critical success factors of agile methodologies. The ethical considerations included making sure everything was ready before the respondents arrived, informing the respondents about the purpose of the study, discussing the benefits and risks of the study, as well as the measures and the obligations of both the respondents and the researchers.

Potential participants were informed that they had the right to decide whether or not they wished to participate in the study.

4.7.3 Confidentiality, anonymity and privacy

Privacy refers to the right that all information attained in the course of the study could be safeguarded against invaders. Confidentiality was assured by restricting access to the gathered data to the researcher, statistician and supervisors. Moreover, information obtained from the participants would not be shared without their authorisation. Also, participant anonymity was guaranteed by not using the actual names of participants in the study (Leedy & Ormrod, 2005). The finalised questionnaires will be demolished after twelve years.

4.7.4 Beneficence and justice

Justice necessitates that individuals participating in the research must be treated fairly. The results of the questionnaire did not affect the IT experts, managers and software developers or any other member's lives at their work place. The research will ultimately benefit IT companies and software development companies and did not harm the organisations. Moreover, the researcher did not foresee any potential impairment for the respondents since their identities would not be disclosed. In contrast, it is anticipated that the study will benefit the respondents directly in that it made IT managers and software developers aware of the elements of critical success for software development thereby hopefully reducing the software failure rate in South Africa.

4.8 Summary

This chapter discussed the research approaches that were used to carry out the study. Research approaches in general, and the approach adopted by the study were explained. Various methods that were used to collect and determine the critical success factors were discussed. The themes from the interviews were used to categorise the factors and this is discussed in detail in Appendix E. This chapter also discussed the research questions, conceptual framework and proposed hypotheses, findings of which are explained in chapters 5 and 6. The chapter then discussed the ethical considerations with regard to the study.

CHAPTER 5: DATA ANALYSIS AND HYPOTHESES TESTING

5.1 Introduction

The aim of this chapter is to discuss the statistical results of the study and to integrate the empirical findings with the literature review to answer the research questions. The statistical analysis of the data was carried out with Statistical Package for Social Sciences (SPSS) v23.0 and Analysis of Moment Structures (AMOS) v23.0 and the findings are discussed below, showing the results of the approaches that were used to screen and categorise the critical success factors of agile software projects.

The social and economic characteristics of the participants are explained using descriptive statistics. Data screening was assessed by means of multivariate assumptions through use of multiple regression and residual analysis. The hybrid approach of both confirmatory factor analysis (CFA) and exploratory factor analysis (EFA) was used to test validity since there were no clear guidelines regarding which one was better. The Cronbach alpha and composite reliability test of the constructs was evaluated in the data set survey. The correlation between constructs was evaluated to determine whether there was a significant relationship between them in this study in the South Africa context. This chapter then presents a model of success factors of agile software projects based on actual success, developed by means of a structural equation model (SEM). The chapter concludes with a summary of the results.

5.2 Demographic factors of the respondents

A total of 500 questionnaires were distributed, of which 460 were returned. All the questionnaires returned were suitable for analysis. The demographic information of the 460 participants is shown in Table 5.1. In terms of gender, 296 males responded, accounting for 64.6%, and 164 females responded, accounting for 35.4% of the sample. Most of the participants fell into the age group of 31-35 years (37.8%), followed by the 26-30 age group (24.6%), the 21-25 age group (24.6%), and the 40 and above age group (7.4%), with those younger than 20 years accounting for 5.7% of the sample. The majority of the participants (193) were employees with a Degree qualification (42.0%), followed by those with a Diploma qualification (28.0%), a Certificate qualification (21.3%), a Bachelors of Technology (B Tech) or Bachelor Degree of Science Honours (Bachelor of Honours) qualification (5.9%), and a Master of Science qualification (MSc) or PhD qualification (2.8%).

Table 5.1: General demographic attributes of the respondents (N = 460)

Demographic factors	Frequency	Percentage
Gender		
Male	296	64.6
Female	164	35.4
Age		
Less than 20	26	5.7
21-25	113	24.6
26-30	113	24.6
31-35	174	37.8
40 and above	34	7.4
Education		
Certificate	98	21.3
Diploma	129	28.0
Degree	193	42.0
Bachelor of Honours and B Tech	27	5.9
MSc and PhD	13	2.8

Other demographic factors were categorised into three groups, namely project type, company profile and respondent profile, as discussed below.

5.2.1 Project type

The projects type describes the project activities of the respondents' daily business operations, with information provided relating to its area of application, its contents and its platform. Agile methodologies used, project scope and project location are shown in Table 5.2 below.

5.2.1.1 Agile methodologies used

There are several agile methodologies mentioned in the literature review but only the four most commonly used in South Africa were analysed from the data collected, namely Extreme Programming (XP), Scrum, Feature Driven Development (FDD) and Lean Software Development (LSD). The other three (Crystal Development, Dynamic System Development method and Adaptive Software Development) are not included in the study results because no respondents selected these options. Table 5.2 shows the distribution of projects, showing XP (38.9%) and

Scrum (34.6%) as the most widely used agile methodologies (jointly making up almost three quarters of all projects in the study), followed by a FDD (13.9%) and then Lean Software Development (12.6%).

5.2.1.2 Project scope

As for the project scope, the demographic aspect considers the size of the agile project with regards to the number of team members as well as the length of the project. Table 5.2 shows that most projects have between 2 and 5 team members (79.1 %), which confirmed the assumption that most agile software projects have small team sizes, such as pair programming in XP. Moreover, 20.9 % of the sample had slightly larger teams (6-10 team members), indicating that agile software projects could involve a larger number of people working together, for example in Scrum which favours eight or fewer individuals as stated by Dyba and Dingsoyr (2009).

5.2.1.3 Project length

As for the length of the project, the longest projects were 13-18 months long, making up 52.0% of the sample, while the shortest were between 6-12 months, making up 23.9% of the sample. This demonstrates that agile software projects have fast delivery rates, as expected.

5.2.1.4 Project location

As for the project location, the demographic data gives insight into how agile software project practice differs across geographical regions. This includes the location where the main project took place (province) and the corresponding zone (South Africa). Participants were asked to select a province from a pre-defined pull-down list. Table 5.2 shows that out of 8 provinces reported, Gauteng had the highest number of projects (31.3 %), followed by the Western Cape (21.1 %), Mpumalanga (8.3%), Kwa-Zulu Natal (8.0%), Eastern Cape (7.8%), Northern Cape (7.8%), Free State (7.8%), and Limpopo (7.8%). Gauteng and the Western Cape account for over 50% of all projects reported, with the remaining six provinces accounting for the rest.

Table 5.2: Demographic attributes of the respondents (Project type) (N = 460)

Demographic factors	**Frequency**	**Percentage**
Agile Methodologies		
Extreme Programming	179	38.9
Scrum	159	34.6
Feature Driven Development	64	13.9
Lean Software Development	58	12.6
Team Members		
2-5 team members	364	79.1
6-10 team members	96	20.9
Province		
Gauteng	144	31.3
Western Cape	97	21.1
Mpumalanga	38	8.3
Eastern Cape	36	7.8
Northern Cape	36	7.8
Free State	36	7.8
Limpopo	36	7.8
Kwa-Zulu Natal	37	8.0
Length Of the Project		
6-12 months	110	23.9
13-18 months	239	52.0
More than 18 months	111	24.1

5.2.2 Company profile

Demographic data on the company profile includes information relating to the sectors in which projects are developed (namely ICT, software development, IT, banking, manufacturing and education) as well as the, company size and company revenue. The company profile also provides organisational information with the regards to agile software projects.

5.2.2.1 Company's industry

Table 5.3 below shows that software development companies (30.9%) were the most popular type, followed by ICT (22.2%), manufacturing (20.7%), IT (13.3%), banking (11.5%) and then education (1.5%). This shows that agile software development practices are being spearheaded by software development companies for their own projects or for their customers' projects, while the rest were in-house projects carried out by manufacturing, ICT, IT, banking and education departments of non-software companies.

5.2.2.2 Company size and revenue

The size of the company was measured by the annual revenues and number of the employees. Table 5.3 shows that agile software projects were likely to occur in small organisation with 64.3 % of the projects in organisations of 1-20 employees, followed by 21-40 employees (15.9%), 41-60 employees (7.0%) and then 61-80 employees (9.3%). In addition, a few large organisations were involved in agile software project practice: 3.5% of the projects were carried out by organisations having over 80 employees. Further analysis of the industry results indicates that an extensive variety of industries use agile methodologies and practices.

Company revenue indicates that agile methodologies were adopted mostly in companies earning more than R250 000 000 per year (25.7%), followed by R50 000 001- R100 000 000 (24.6%), R150 000 001-R200 000 000 (24.1%), R10 000 001-R50 000 000 (18.9%), R0-R10 000 000 (3.5%), and then R100 000 001-R150 000 000 (3.3%). This shows that some companies with 1-20 employees have a high revenue of more than R250 000 000 per year. This indicates that agile software projects with a smaller group of team members can produce high productivity.

Table 5.3: Demographic attributes of the respondents (Company Profile) (N = 460)

Demographic factors	Frequency	Percentage
Company industry		
ICT	102	22.2
Software Development	142	30.9
IT	61	13.3
Bank	53	11.5
Manufacturing	95	20.7
Education	7	1.5
Company size		
1-20 employees	296	64.3
21-40 employees	73	15.9
41-60 employees	32	7.0
61-80 employees	43	9.3
More than 80 employees	16	3.5
Company revenue (in Rands)		
0-10 000 000	16	3.5
10 000 001-50 000 000	87	18.9
50 000 001-100 000 000	113	24.6
100 000 001-150 000 000	15	3.3
150 000 001-200 000 000	111	24.1
More than 250 000 000	118	25.7

5.2.3 Respondent profile

The respondent's demographic profile could provide background information on each respondent's viewpoint such as their job responsibility within the project at hand, as well as their level of experience with agile software projects, including the number of agile software projects they had been involved in.

5.2.3.1 Job responsibility

Table 5.4 below indicates the breakdown of job responsibility within agile software projects of all the participants. The top two job positions represent 87 % of the sample population namely project team leaders (62%) and project managers (25%), followed by organisation management (8.5%) and then team members (4.6%). This means, as far as job responsibility is concerned, that around

92% of the participants were intimately involved with agile software projects. Most of the participants (50.0%) fell into the experience group of 3-4 years, followed by the 0-2 years (25.0%) and 5-6 years (25.0%). The experience results concur with the notion that agile software development project teams are not particularly dependent on experienced participants. A counter argument though is that these people may have experience in other methodologies, such as spiral, prototyping and the waterfall model, or have been involved in software engineering projects for long enough to understand the complexities related to their projects.

Table 5.4: Demographic attributes of the respondents (Respondent profile) (N = 460)

Demographic factors	Frequency	Percentage
Job position		
Project manager	115	25.0
Team leader	285	62.0
Team member	21	4.6
Organisation Management	39	8.5
Experience (in years)		
0-2	115	25.0
3-4	230	50.0
5-6	115	25.0
Total	460	100.0

5.3 Principal component analysis

5.3.1 Bartlett's test of sphericity and the Kaiser-Meyer-Olkin value

Tables 5.5 and 5.6 depict the results of Bartlett's test of sphericity, the Kaiser-Meyer-Olkin (KMO) value and the communalities. The KMO value of 0.803 is the degree of common variance among the nine variables and is significant enough to conduct a factor analysis (KMO > 0.6) (Pallant, 2013). The p-value of Bartlett's test (p = 0.000), which is below 0.01, is significant at the 99% confidence level and thus suitable to factor analysis. This result indicates that the correlation structure is significantly strong enough to perform a factor analysis of the items.

Table 5.5: KMO and Bartlett's Test

KMO and Bartlett's Test		
Kaiser-Meyer-Olkin Measure of Sampling Adequacy.		0.911
Bartlett's Test of Sphericity	Approx. Chi-Square	17672.24
	d.f	1081
	Sig.	0.000

Source: SPSS 23.0

5.3.2 Communalities

The communalities indicate the extent to which an individual item correlates with the other items (Hartung & Knapp, 2005; Pallant, 2013). A value close to 1 indicates an item that correlates highly with the rest of the items. Items with low communalities (near 0.3) should be eliminated (Pallant, 2013; Rohani, Yusof & Mohamad, 2009).

Using the Maximum Likelihood Analysis method of extraction (discussed in detail below), the communalities for all 47 items in Table 5.6 are observed to be reasonable.

Table 5.6: Extraction of communalities

Communalities			**Communalities**		
Questions	**Initial**	**Extraction**	**Questions**	**Initial**	**Extraction**
Process_1	0.661	0.663	Intention_1	0.747	0.705
Process_2	0.737	0.751	Intention_2	0.825	0.802
Process_3	0.781	0.824	Intention_3	0.857	0.861
Process_4	0.746	0.761	Intention_4	0.879	0.897
Process_5	0.733	0.743	Intention_5	0.753	0.711
Project_1	0.673	0.816	Intention_6	0.681	0.589
Project_2	0.631	0.694	Intention_7	0.64	0.552
Project_3	0.57	0.609	Tech_1	0.46	0.353
Effort_1	0.628	0.596	Tech_2	0.66	0.728
Effort_2	0.79	0.766	Tech_3	0.511	0.389
Effort_3	0.75	0.738	Tech_4	0.677	0.825
Effort_4	0.807	0.783	Actual_1	0.541	0.436
Effort_5	0.741	0.728	Actual_2	0.737	0.712
Effort_6	0.77	0.708	Actual_3	0.653	0.586

Communalities			Communalities		
Effort_7	0.815	0.779	Actual_4	0.64	0.592
People_1	0.715	0.768	Actual_5	0.771	0.767
People_2	0.764	0.806	Actual_6	0.787	0.804
People_3	0.722	0.73	Actual_7	0.53	0.476
People_4	0.634	0.609	Org_1	0.629	0.562
Performance_1	0.632	0.637	Org_2	0.763	0.76
Performance_2	0.605	0.589	Org_3	0.816	0.895
Performance_3	0.595	0.611	Org_4	0.676	0.589
Performance_4	0.597	0.603	Orgl_5	0.699	0.649
			Org_7	0.586	0.555

Extraction Method: Maximum Likelihood (MLA).

5.3.3 Maximum likelihood analysis

Exploratory factor analysis (EFA) was used to analyse the questionnaires based on the Maximum Likelihood Analysis (MLA) technique through use of the Promax Rotation Method (PRM) to examine distinctions among the constructs. The main objective of using MLA was to reduce the factors to the small set of composite variables, by pasting the results into the pattern matrix plug in in AMOS 23.0 from SPSS 23.0 (Pallant, 2013). In addition, EFA factor analysis was used to identify the hidden dimensions or constructs which may or may not be apparent from direct analysis (Pallant, 2013). The conceptualisation of the questionnaires was based on reviewing related literature on agile software projects. The nine categories that had been used to group the factors were also established based on the literature review and interviews. Furthermore, it was important to check whether all the factors were loading in their categories and that their eigenvalues were reasonable enough to be included in the final analysis (Pallant, 2013).

Cultural, social and political constructs were excluded because these were found to have lower loadings, and eigenvalues less than 1.0. Table 5.7 shows an extract from the component matrix, with the results from the total variance explained. Table 5.7 illustrates the loadings for each factor on each rotated component that forms the categories of the classification. The results consist of nine factors that have eigenvalues greater than 1.0.

A further analysis was carried out using the scree plot. Figure 5.1 below shows the scree plot results. The scree plot does not give a clear idea of which factors to retain. As an alternative a couple of rules of thumb need to be followed. For instance, one rule is to regard only those with eigenvalues above 1. All the eigenvalues were plotted in their decreasing order. Taking into account the different criteria, the decision was made to extract 9 factors as shown Figure 5.1.

Table 5.7: Results of maximum likelihood analysis of critical success factors of agile software projects

Category	**Success factors**	**Rotated component matrix value**	**Total variance explained**	**Eigenvalues**
Effort expectancy	Effort expectancy (1)	0.726	27.252	3.335
	Effort expectancy (2)	0.854		
	Effort expectancy (3)	0.868		
	Effort expectancy (4)	0.893		
	Effort expectancy (5)	0.857		
	Effort expectancy (6)	0.807		
	Effort expectancy (7)	0.888		
Intention factors	Intention behaviour (1)	0.828	9.465	2.37
	Intention behaviour (2)	0.831		
	Intention behaviour (3)	0.937		
	Intention behaviour (4)	0.956		
	Intention behaviour (5)	0.874		
	Intention behaviour (6)	0.733		
	Intention behaviour (7)	0.672		
Organisational factors	Organisation (1)		7.626	4.761
	Organisation (2)	0.609		
	Organisation (3)	0.905		
	Organisation (4)	0.944		
	Organisation (5)	0.72		
	Organisation (7)	0.725		
Actual success	Actual success (1)	0.629	6.556	2.495
	Actual success (2)	0.846		
	Actual success (3)	0.708		

Category	Success factors	Rotated component matrix value	Total variance explained	Eigenvalues
factors	Actual success (4)	0.671		
	Actual success (5)	0.935		
	Actual success (6)	0.94		
	Actual success (7)	0.582		
Process factors	Process (1)	0.771	4.649	13.264
	Process (2)	0.872		
	Process (3)	0.914		
	Process (4)	0.862		
	Process (5)	0.866		
People factors	People (1)	0.902	4.313	1.637
	People (2)	0.893		
	People (3)	0.836		
	People (4)	0.727		
Performance expectancy	Performance expectancy (1)	0.689	1.735	3.93
	Performance expectancy (2)	0.665		
	Performance expectancy (3)	0.651		
	Performance expectancy (4)	0.603		
Project factors	Project (1)	0.908	2.933	2.07
	Project (2)	0.795		
	Project (3)	0.753		
Technological factors	Technological (1)	0.542	3.784	1.087
	Technological (2)	0.849		
	Technological (3)	0.547		
	Technological (4)	0.914		

Extraction method: Maximum Likelihood.
Rotation method: Promax with Kaiser Normalisation.
Rotation converged: in 7 iterations.

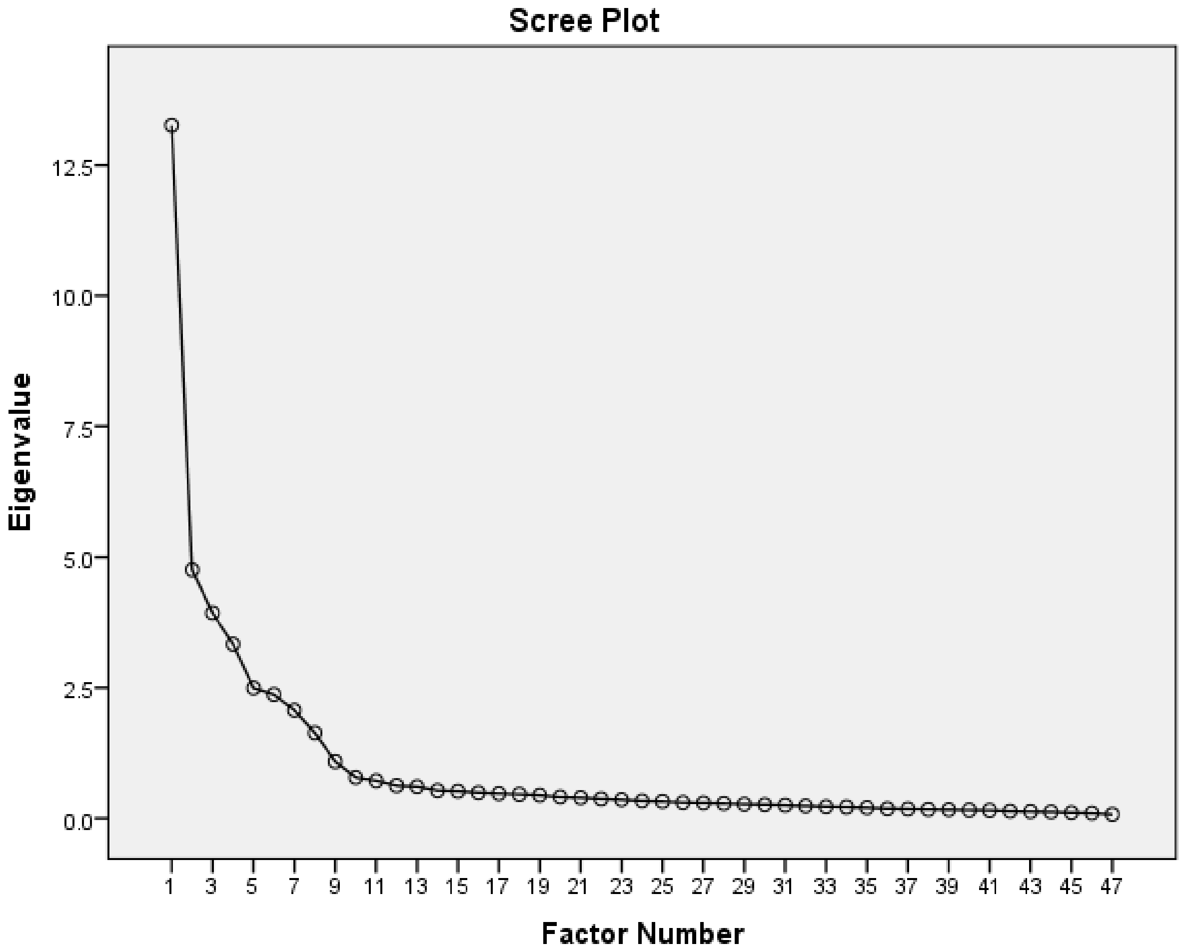


Figure 5.1: South Africa data sample scree plot

5.3.4 Interpretations of MLA results

According to Schreiber (2008), the rotation method assists in the identification of factors that load in each category. Normally a factor is said to be loaded in a category if its loading value is greater than 0.3 (Pallant, 2013). Table 5.7 shows that the first category extracted (effort expectancy factors) has the highest total variance explained. This implies that the effort expectancy category accounts for the most variance of the observed factors, meaning it is well correlated with many of the observed factors. When this statistical observation is reflected back to this study, it implies that effort expectancy plays a greater role in agile software projects than the rest of other factors. On the other hand, the next extracted category explains the highest variance of those factors that were not accounted for by the first category. This implies that this category correlates with many of those factors that were less or not very much correlated with the first category. On the same note, this study shows that intention factors strongly influence agile software projects but many of these factors are independent of effort expectancy factors and their correlation may tend to zero.

The analysis showed that some factors had split loadings, implying that these factors loaded in more than one category. These factors included Org_6, Org_8, Tech_5, Tech_6, Tech_7, Project_4, Project_5, Project_6 and Project_7 which loaded variously in the organisation, technological and project categories. Furthermore, political, social influence and cultural influence were excluded by the exploratory factor analysis, as mentioned earlier. Therefore, their loading difference was higher in their initial categories, and their complex loadings were removed. The MLA of the 47 factors output and 7 iterations of extractions conformed to the nine categories which had earlier been suggested by the literature review and the interviews carried out. In the end, the initial 72 factors were reduced to 47 factors that make a total contribution of 68.312% of the total variance. The extraction Table 5.7 further shows the factors in the effort expectancy category making the highest, and technological categories making the lowest contributions of the research model, with a contribution of 27.252% and 1.735% respectively of the total variance explained. This adds to the call made by several researchers that validation of agile software projects' critical success factors should be done in countries with low technology development to check whether the established projects' critical success factors have the same influence in these countries as in high technology developed countries.

5.4 Data screening

Data screening was carried out to check for missing data, accuracy of data entry, normality and miscoded data. In order to avoid normality and linearity problems, missing values and outer influences that could have improved the R squared values are highlighted. Data screening was carried out to improve the data so that the statistical analysis procedures are precise and to ensure that estimates have a sound basis. Appropriateness of data entry was taken note of during the filling of the questionnaire or during the capturing of the data on the system. To get rid of biased results, it was vital to check all the data sets for cases of univariate and multivariate outliers and, where they were found, to make a meaningful decision. It was also vital to check for the normality of the data so as to discover what to do with cases of non-normality.

Three procedures were carried out to avoid the circumstances of missing data. The researcher ensured that no questionnaire was captured before checking whether it was fully completed. All questionnaires with missing data were discarded. In addition, physical proof analysis of the data that had been entered in SPSS v 23.0 was done against the original data on the questionnaires. Data proof analysis was carried out in two stages. After entering the data on the SPSS v 23.0 software, the investigator did an item by item check to confirm that all data had been correctly captured. Furthermore, an independent assistant was asked to check for missing data. Microsoft Office and Excel v 2016, SPSS v 23.0 and AMOS v 23.0 were used to check for missing data for the questionnaire.

5.4.1 The South African data sample

Pallant (2013) recommended a linear model that must be tested with diagnostic plots to confirm validity of the assumptions of multiple regression and residual analysis. In addition, Dion (2008) and Pallant (2013) suggested that several assumptions must be taken note of to lead to a meaningful statistical analysis of the data. The assumptions includes:

- analysing for linear functional form,
- having a proper representative sample,
- determination of fixed independent factors or variables and observations of the model to confirm that there are no omitted factors,
- equality of variance of the errors that provides homogeneity of residual variance,
- normality of the residuals or errors,
- checking whether data is absent form multicollinearity and homoscedasticity,
- removing high correlation of the errors, and
- ensuring that outliers are noticed and removed (Dion, 2008; French & Finch, 2006; Hartung & Knapp, 2005; Pallant, 2013).

Singularity and multicollinearity are associated to the correlation matrix and they happen when decision variable are auto-correlated with values of 0.9 and above (Pallant, 2013). Multivariate and bivariate correlations were analysed and no bivariate correlations of 0.9 and above were found between the independent variables. Multivariate correlation was evaluated through the coefficients output and the residual analysis. In addition, all tolerance readings were above 0.3 and the variance inflation factors were below 3.

Parameters in SPSS were used to detect the outliers through comparison of the values of the residuals (error = predicted – actual) as being out of the range of 3.5 and -3.5 of standardized residuals (Hartung & Knapp, 2005; Pallant, 2013).

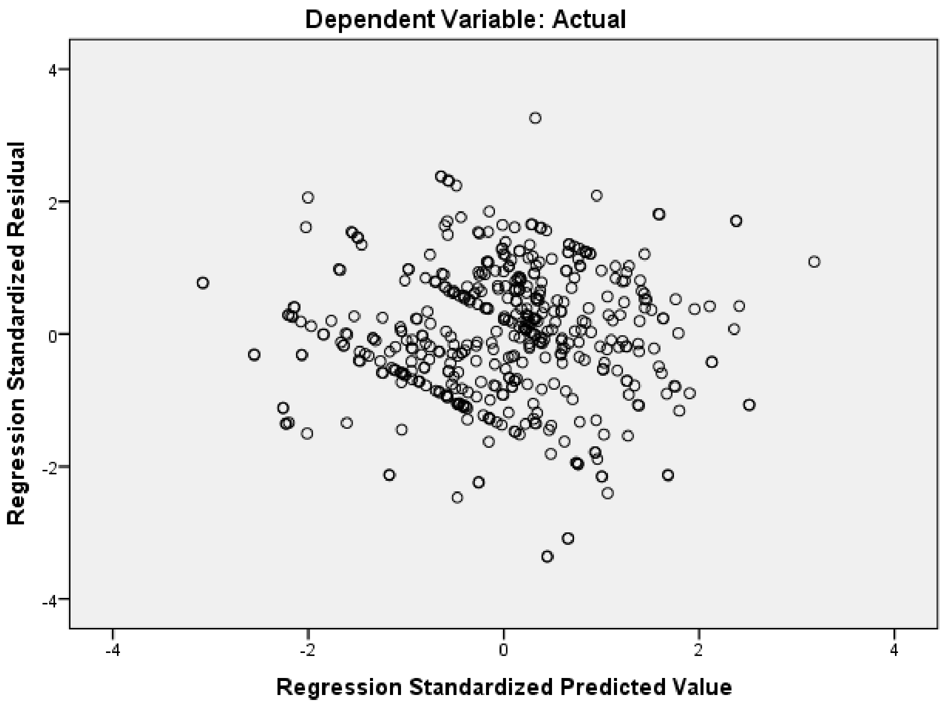


Figure 5.2: South Africa dataset to diagnostic test with scatter plot

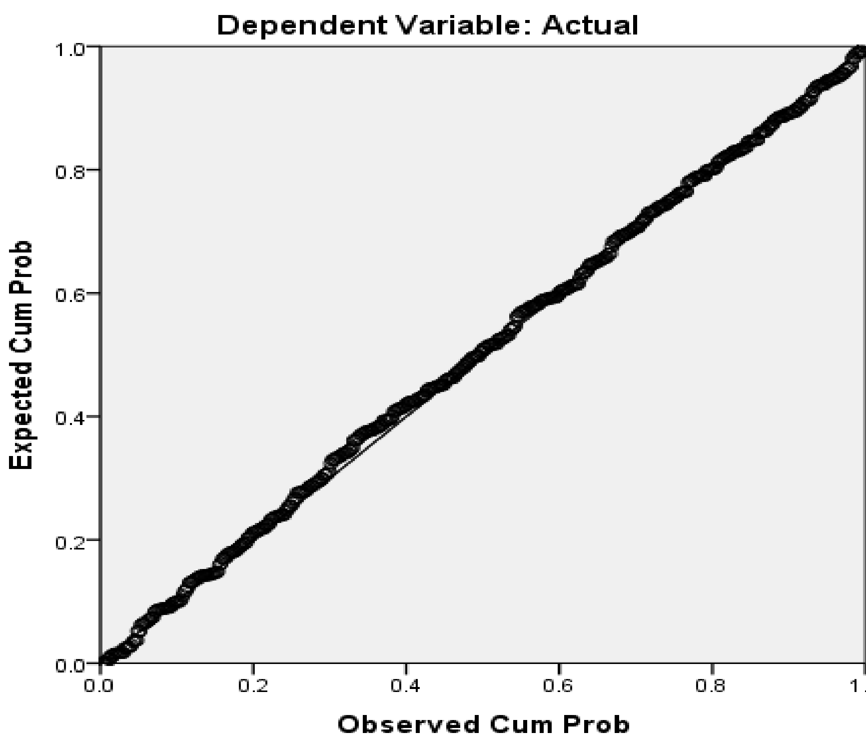


Figure 5.3: South Africa dataset for a normal P-P Plot of regression standardised residual

Regression was run in SPSS v 23.0 to determine the residual versus the fitted plot to find out which errors were out of range (as shown in Figure 5.2). Mahalanobis distance measure of the geometric distance between a given point on the graph and the centroid was used to detect the univariate and multivariate outliers through use of recommended assumptions (Barrett, 2007; Hartung & Knapp, 2005). Multivariate and univariate outliers were examined using the residual analysis and no univariate and multivariate outlier cases with a residual above 3.5 were detected. It was concluded that the multivariate outliers were random and there was little danger in retaining the factors. From Figure 5.3, the residuals indicate linearity, homoscedasticity and independency since the scores seem as if they were organised between two parallel lines.

Multicollinearity (which describes the high correlation among the explanatory decision variables that prevent their effects from being analysed) was tested (Cheung & Rensvold, 2002). Hartung and Knapp (2005) suggested that the presence of multicollinearity makes it problematic to evaluate the influence of unidentified parameters giving significant errors in minor changes in data. Thus, this will lead to high significant standard errors with a high correlation coefficient that generates a value of R square close to 1 or -1 (Byrne, 2013). This was examined by comparing the R square to verify whether it is near ±1. The modal summary of SPSS produced a value of R square of 0.337. When this value was computed for tolerance (tolerance = 1- R^2 = 0.663) – based on the criterion that deletion would not improve the regression R square – the resulting values showed non-existence of multicollinearity.

In addition, the descriptive analysis discovered that the data kurtosis and skewness were within the acceptable value of ±1 therefore no data transformation was needed. Furthermore, the expected normal probability plot was employed to assess multivariate normality. The normal P-P plot of the regression standardised residual was considered normal. Figure 5.3 shows that the scores are almost on the straight line. The plotted scores appear to be closer to the diagonal line, hence scattering is regarded to be normally distributed (Fabrigar, Porter & Norris, 2010). Thus the null hypothesis (that declares that errors follow a normal distribution) is accepted (supported). Therefore, the assumptions for multiple regression and residual analysis are supported (Byrne et al., 2007).

5.5 Reliability, validity of the constructs and correlation

This study used discriminant and convergent validity to determine the degree of correlation between the decision variables and other measures that have been predicted in theory to correlate with them (Hartung & Knapp, 2005). This also aimed to determine whether these decision variables do not in actual fact correlate with other variables that have been theorised not to correlate with them (Pallant, 2013). Decision variables' reliability was also measured using Cronbach's Alpha.

5.5.1 Reliability of the constructs

Before testing for discriminant convergent validity and correlation, the decision variables were evaluated for their reliability. Cronbach's alpha (α) was used to check the internal consistency of the decision variables (Pallant, 2013). Culture, social influence and political decision variables were found to be not reliable ($\alpha < 0.30$) and were therefore excluded from further research and are not shown in Tables 5.8 or 5.9. The remaining α values of the decision variables which were retained were above 0.7, meaning that their corrected item-total correlation is significant, as shown in Table 5.8 (Pallant, 2001 as cited by Chiyangwa, and Alexander, 2016). The findings indicate that the Cronbach's alpha values of all items range from 0.819 to 0.948, with an overall internal consistency reliability of 0.920 for 47 items. Organisational factors indicated that if Org6 is deleted its reliability will improve to 0.813. In addition, organisation factors indicated that a deletion of Org8 would further improve the reliability to 0.915. These results concurred with the findings attained from the testing of the model fit which suggested the same variables should be deleted. This implied that these two variables should not be considered for further analysis. However, since AMOS v 23.0 does not have a Wald test function to detect whether deleting a parameter would increase or reduce its fitness, caution was taken when this was done (Rohani et al., 2009). It was therefore vital to use the delete and test method so as to avoid making the model unidentifiable and also to avoid deviation between the structural and the measurement model.

Table 5.8: Cronbach's alpha reliability of the constructs for each data sample

Scale Items	Mean	SD	Corrected item-total correlation	Cronbach alpha (α) if Item deleted	Total Cronbach α
Effort expectancy factors					0.947
Effort_1	3.43	0.962	0.747	0.945	
Effort_2	3.48	0.967	0.844	0.937	
Effort_3	3.53	0.953	0.833	0.938	
Effort_4	3.70	0.891	0.852	0.936	
Effort_5	3.34	1.004	0.823	0.939	
Effort_6	3.62	0.915	0.807	0.94	
Effort_7	3.69	0.931	0.852	0.936	
Intention factors					0.948
Intention_1	4.03	0.748	0.816	0.94	
Intention_2	3.97	0.746	0.853	0.937	
Intention_3	3.97	0.745	0.867	0.936	
Intention_4	3.96	0.727	0.886	0.934	
Intention_5	4.05	0.726	0.839	0.938	
Intention_6	4.06	0.716	0.77	0.944	
Intention_7	4.08	0.725	0.733	0.947	
Organisational factors					0.915
Org_1	3.76	0.781	0.678	0.911	
Org_2	3.87	0.775	0.796	0.894	
Org_3	3.88	0.787	0.857	0.885	
Org_4	3.72	0.798	0.735	0.903	
Orgl_5	3.75	0.744	0.798	0.894	
Org_7	3.82	0.766	0.698	0.908	
Actual factors					0.915
Actual_1	2.72	0.85	0.642	0.912	
Actual_2	2.82	0.909	0.827	0.893	
Actual_3	2.68	0.896	0.734	0.903	
Actual_4	3.04	0.953	0.73	0.903	
Actual_5	2.97	0.974	0.784	0.897	
Actual_6	2.99	0.978	0.82	0.893	

Scale Items	Mean	SD	Corrected item-total correlation	Cronbach alpha (α) if Item deleted	Total Cronbach α
Actual_7	2.86	0.925	0.641	0.912	
Process factors					0.934
Process_1	2.37	0.939	0.777	0.927	
Process_2	2.33	0.994	0.827	0.918	
Process_3	2.19	0.85	0.864	0.912	
Process_4	2.23	0.918	0.836	0.916	
Process_5	2.22	0.858	0.824	0.919	
People factors					0.909
People_1	3.38	1.075	0.8	0.88	
People_2	3.3	1.121	0.83	0.869	
People_3	3.5	1.038	0.817	0.874	
People_4	3.67	0.99	0.732	0.903	
Technological factors					0.819
Tech_1	1.42	0.896	0.559	0.808	
Tech_2	2.05	1.128	0.692	0.748	
Tech_3	1.53	0.932	0.6	0.791	
Tech_4	2.11	1.14	0.732	0.727	
Project factors					0.870
Project_1	3.09	1.135	0.781	0.791	
Project_2	3.23	1.075	0.763	0.808	
Project_3	2.89	1.102	0.714	0.851	
Performance expectancy factors					0.859
Performance_1	4	0.67	0.727	0.81	
Performance_2	3.86	0.721	0.703	0.821	
Performance_3	4.12	0.627	0.707	0.82	
Performance_4	3.93	0.718	0.684	0.829	

Number of participants (N) = 460

5.5.2 Convergent and discriminant validity and correlation

Convergent validity defines the proportion of variance for all factors that are related (Al Tamimi, 2014; Hair, Black, Anderson & Tatham, 1995; Pomykalski, Dion & Brock, 2008). Convergent validity is determined from the computation of the composite or construct reliability (CR) and variance extracted (VE) for each indicator of the construct. As recommended by Hartung and Knapp (2005) and Dion (2008), factor loadings, composite reliability and the average variance extracted (AVE) were used to assess the convergent validities. The discriminant validity was considered by examining whether or not the square root of AVE exceeded the correlations between constructs. The reliability was evaluated by examining the internal consistency reliability as recommended by (Venkatesh et al., 2003).

SPSS v 23.0 and Amos v 23.0 were used and the generated results are summarised in Tables 5.9 and 5.10 and Figure 6.3. As shown in Table 5.10, all factors in the measurement model have acceptable composite reliability and convergent validity: the indicators' standardised loadings [λ]) are significant ($p<0.001$), the composite or construct reliabilities exceed the acceptable criteria of 0.7, and the AVEs were greater than the threshold value of 0.5 in all cases (Fabrigar et al., 2010; French & Finch, 2006).

In Table 5.10 the diagonal elements are the square roots of AVE, and off-diagonal elements are correlations between decision variable (constructs). The Table illustrates that the discriminant validity and reliability are supported since all diagonal elements are higher than the off-diagonal elements in the corresponding columns and rows, and internal consistency reliability is above 0.7.

As recommended by Byrne (2013), and Dion (2008), the average variance extracted and the composite reliability can consequently be derived statistically from the factor loadings and the measurement errors. Equations (4) to (6) represent the scientific association between the measuring terms below.

(4) Error term = $\epsilon = 1- \lambda^2$

(5) Composite reliability = C. R. = $[\Sigma 1 \ldots n (\lambda)]^2 / ([\Sigma 1 \ldots n (\lambda)]^2 + \Sigma 1 \ldots n (\epsilon))$

(6) Average variance extracted=AVE = $[\Sigma 1 \ldots n (\lambda)^2] / ([\Sigma 1 \ldots n (\lambda^2)] + \Sigma 1 \ldots n (\epsilon))$

Where λ is the standardised regression weight for the indicators, n is the number of indicators for each decision variable and ϵ is the error term.

By means of the correlation and standard regression weights (which are the standardised loadings), the average variance extracted and composite reliability are attained. The three equations were run in Stats Tools Package 2016 and AMOS v 23.0, and the extracts are shown in Table 5.10.

Table 5.9: Standardised loadings, composite reliability and average variance extracted

Variable constructs	**Indicators**	**Standardised loadings λ**	**Composite reliability (C.R.)**	**Average variance extracted (AVE)**
Effort expectancy factors	Effort_1	.779	0.935	0.705
	Effort_2	.881		
	Effort_3	.867		
	Effort_4	.890		
	Effort_5	.843		
	Effort_6	.805		
	Effort_7	.852		
Intention behaviour factors	Intention_1	.792	0.942	0.731
	Intention_2	.875		
	Intention_3	.935		
	Intention_4	.954		
	Intention_5	.827		
	Intention_6	.746		
	Intention_7	.708		
Organisational factors	Org_1	.753	0.892	0.624
	Org_2	.793		
	Org_3	.849		
	Org_4	.816		
	Orgl_5	.851		
	Org_7	.742		
Actual behaviour factors	Actual_1	.699	0.905	0.656
	Actual_2	.879		
	Actual_3	.796		
	Actual_4	.777		
	Actual_5	.776		

Variable constructs	**Indicators**	**Standardised loadings λ**	**Composite reliability (C.R.)**	**Average variance extracted (AVE)**
	Actual_6	.810		
	Actual_7	.667		
Process factors	Process_1	.807	0.935	0.743
	Process_2	.861		
	Process_3	.904		
	Process_4	.875		
	Process_5	.860		
People factors	People_1	.871	0.902	0.698
	People_2	.924		
	People_3	.815		
	People_4	.716		
Technological factors	Tech_1	.545	0.880	0.786
	Tech_2	.857		
	Tech_3	.545		
	Tech_4	.913		
Project factors	Project_1	.869	0.872	0.694
	Project_2	.853		
	Project_3	.775		
Performance expectancy factors	Performance_1	.802	0.860	0.607
	Performance_2	.763		
	Performance_3	.783		
	Performance_4	.767		

South Africa data sample

(Number of participants (N) =460)

Table 5.10: The square of AVEs (shown in bold at diagonal) and factor correlation coefficients

	Project	**Effort**	**Intention**	**Organisation**	**Actual**	**Process**	**People**	**Technological**	**Performance**
Project	0.833								
Effort expectancy	0.022	0.84							
Intention	0.047	0.497***	0.855						
Organisation	0.104**	0.428***	0.561***	0.79					
Actual	-0.014	0.543***	0.39***	0.338***	0.81				
Process	-0.068	0.203***	0.228***	0.212***	0.375***	0.862			
People	0.376***	0.030*	0.066	0.065	0.011	-0.096	0.835		
Technological	0.261***	0.094	-0.035	-0.021	-0.005	-0.068	0.248***	0.886	
Performance	0.044	0.531	0.549***	0.764***	0.404***	0.195***	0.111**	0.03	0.779

**p<0.05 **p<0.01 ***p<0.001*

Note: Effort means Effort expectancy factors, and Performance means Performance expectancy factors.

As seen in Tables 5.9 and 5.10, the values of composite reliability for each latent decision variable are in the range of 0.860 to 0.942. The standardised loadings are all above 0.5 for all measuring items, ranging from 0.545 to 0.954. Likewise, the average variance extracted values for each latent factor are in the range 0.607 to 0.786 and are thus above the lower limit of 0.5 (Byrne, 2013).

5.6 Valuation of the measurement model and confirmatory factor analysis

After analysing the descriptive statistics, multivariate, validity, reliability and correlation of the sample data, this study used structural equation modelling (SEM) to test the research model. SEM was an appropriate method for explaining the latent behaviour that leads to the success factors of agile software projects (Byrne et al., 2007). Likewise, SEM was expected to give a favourable consideration of the relationship between these latent independent variables and the indicators that measure the research model. It is also important to note that SEM uses cross-sectional variation when modelling the framework (Dion, 2008). SEM was expected to model the variations of respondents in this study since the aim was to validate the model across agile professionals and agile management staff with different education, gender, age and levels of experience in South Africa.

Confirmatory factor analysis (CFA) was used to reduce the measurement error. Since some errors might arise due to misspecifications, a model fit was used to allow the redefinition of the model. This study is in line with Byrne and Stewart's (2006) suggestions that when using SEM for data analysis six steps ought to be followed, namely: developing individual decision variables, developing the overall measurement model, designing a study to produce empirical results, specifying the structure model, assessing the measurement model validity and assessing structural model validity. In this study model, structural equation modelling is seen as a combination of confirmatory factor analysis and path analysis.

SPSS v 23.0 and AMOS graphics v23.0 were used to draw the measurement model (Byrne, 2013).

As per Barrett's (2007) recommendations regarding designing the measurement model, a distinction was made between independent variables and dependent variables. In addition, it is vital to differentiate between the observable and latent variables. Latent variables are the decision variables or factors (constructs) which are unobserved but can only be measured by their individual indicators. Latent variables are symbolised by circular or oval shapes on the model while the observable variables are indicated by rectangular or square shapes. The decision variable formed the latent variables while their indicators formed the observable. An error term was attached to each observable or dependent variable (see Figure 5.4). As shown in Figure 5.4, latent variables are associated with each other by means of two headed arrows that represent the covariance between the decision variables (Dion, 2008). The observed variables for each decision variable are linked to each other by one-headed arrows that represent the causal path from the decision variable to the indicator (Byrne et al., 2007).

Exploratory factor analysis (EFA) was used to show that the indicators for each decision variable are a good estimate or measure of the corresponding latent variable(s) (Cohen, Cohen, West & Aiken, 2013; Pallant, 2013). Different studies recommended further analysis using EFA, even though the measurement model has been validated by SPSS v 23.0 software (Byrne et al., 2007; Chen, Sousa & West, 2005; Cheung & Rensvold, 2002). Modification indices were used to improve the model fit when necessary, as recommended by various researchers (Pomykalski et al., 2008; Rohani et al., 2009).

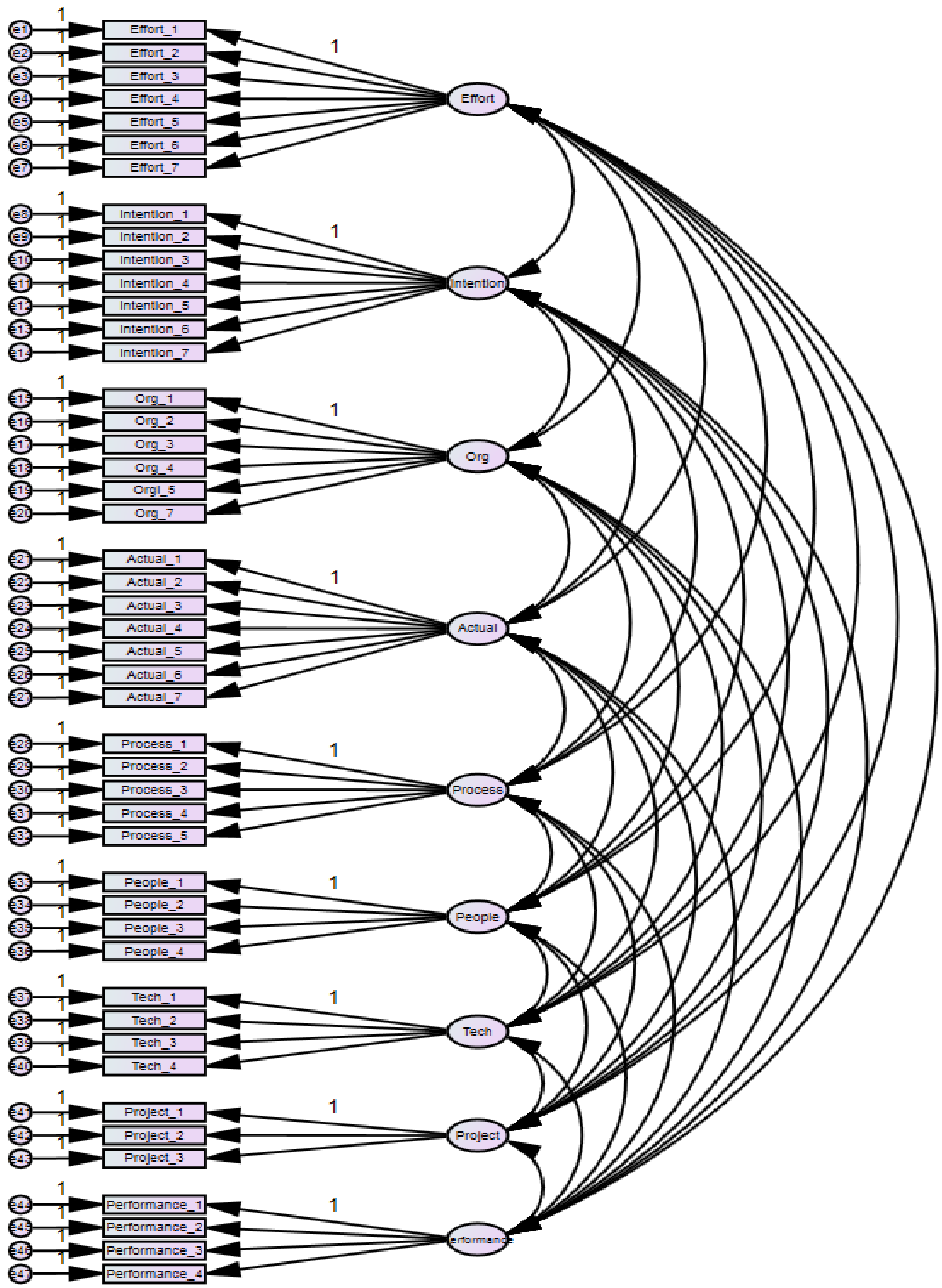


Figure 5.4: Measurement model of critical success factors influencing agile software development projects (Unrefined)

5.6.1 The model fits measurement of CFA

Several researchers have recommended that Chi-square goodness of fit (GOF) is not used as a sole indicator of model fit (Byrne, 2001; Byrne, 2013; Hair et al., 1995). There are numerous chi-square goodness of fit measures which were developed to overcome problems with chi-square sample size. These include the chi-squared (CMIN = χ^2), parsimony, absolute, and incremental fit measures (Yu, 2012). The absolute fit indices are direct measures of how well the proposed model fits the sample data. Instances of such indices include the goodness of fit (GFI) statistic, the root mean square error of approximation (RMSEA), the adjusted goodness of fit index (AGFI) statistic, the root mean square residual (RMR) and standardised root mean square residual (SRMR). Researchers, such as Hair et al. (1995) and Yu (2012), have suggested that the measurement model must be run using the maximum likelihood estimation relying on the data sample (as shown in Figure 5.4). Several researchers – such as Byrne (2001) – state that the relative chi-square must be in the 2:1 range for an acceptable model, others say 3:1 (Pallant, 2013) and others claim relative chi-square less than 1 indicate a poor model fit for the model to be accepted or supported (Yu, 2012).

The model was computed and AMOS v 23.0 showed an output of the optimum threshold with no further error warnings. The common errors in the structural equation model was avoided by constraining the parameter of regression weight to one on each observable variable for each latent variable, as shown in Figure 5.4 (Yu, 2012).

AMOS v 23.0 was computed for a goodness of fit for each model being tested as the default model, for the saturated model which is regarded as the just-identified model and the independence model. The fit of the default model is the one examined in the study and it lies in the limits of the range signified by the saturated and the independence model. Therefore, the assessment of the default model is governed by its position between the limits comparative to the saturate and independence models.

The chi-square (χ^2) test of absolute model fit was χ^2 =2523.167 with 998 degrees of freedom (d.f) and a probability value p= 0.000 suggesting significance at $p < 0.001$. Relying on the value of χ^2 suggests a rejection of the null hypothesis that the model fits the measurement data. This is due to the fact that the null hypothesis for this test is that the model fit the data, hence it is a good model. So a low p-value ($p = 0.000$ where $p < 0.001$) implies that if the null hypothesis is rejected there will be a low likelihood of being mistaken in the conclusions of the research model.

Nevertheless, using chi-square statistics to evaluate model specifications may be problematic in three ways, as discussed below (Cheung & Rensvold, 2002; Dion, 2008; Iacobucci, 2010):

1. The larger the sample size, the more likely the rejection of the model, which can lead to rejecting a good model (This is categorised as a Type II error.)
2. In very large samples, even tiny differences between the observed model and the perfect fit model may be found to be significant.
3. The chi-square fit index is very sensitive to violations of the assumption of multivariate normality.

AMOS v 23.0 produces 25 different goodness-of-fit measures and the choice of which to report is a matter of dispute among researchers (Barrett, 2007; Byrne, 2001; French & Finch, 2006). Dion (2008) and Barrett (2007) suggested that, chi-square χ^2 statistics must be used with at least one incremental index, such as the comparative fit index (CFI), and an absolute index, such as root mean square for error approximation (RMSEA). In addition, experts of research such as Byrne (2001) and Schreiber (2008) have suggested that in case a model has a comparison of varying complexity, the following fit indices can be used: parsimony normed fit index (PNFI) which is a derivation of the normal fit index (NFI), NFI, adjusted goodness of fit (AGFI) or goodness-of-fit index (GFI), and standardised root mean residual (SRMR) or root mean residual (RMR).

Pomykalski et al. (2008) and Dion (2008) suggested the use of several indices to have an acceptable model, namely: SRMR, non-normed fit index (NNFI) also recognised as Tucker-Lewis index (TLI), RMSEA and the comparative fit index (CFI). In addition, researchers such as Rohani et al. (2009) and Mclntosh (2007) strongly supported the ideas of Dion (2008) and Pomykalski et al. (2008) and recommended the use of χ^2 statistics with its degrees of freedom, SRMR, CFI and RMSEA to reach the final model fitness. Byrne (2001) argues that, fit indices are very insensitive to sample sizes, even though some researchers recommended that even minute differences between the observed model and the perfect fit model can be found significant, leading in rejecting something true. This study discusses five fit indices relying on the arguments recommended by Dion (2008), and Cheung and Rensvold (2002) but ultimately follows a combination rule suggested by Pomykalski et al. (2008). The extracted results from the AMOS v 23.0 output of the fit index measurements as compared to their threshold are shown in Table 5.11.

Due to the above issues, Dion (2008) and Byrne (2001) recommended the use of chi-square per degree of freedom (χ^2 / d.f) to make the model insensitive to sample size. The results of the model summary show that CMIN/DF (χ^2 / d.f) = 2523.167/998 = 2.528 which is within the range of the threshold of the ratio of chi-square (χ^2) to the degrees of freedom (d.f) given as 2.1 ≤ (χ^2 / d.f) ≤ 3.1 (as shown in Table 5.11) hence indicating a relatively good fit. This vagueness in the testing of fit models demands the evaluation of more than one fit model to remove the ambiguity (Cheung & Rensvold, 2002).

Table 5.11: Comparison of the model fit Indices with their threshold values (Unrefined)

Fit indices	**Measurement model**	**Threshold**	**Recommendations for the measurement model**
χ^2	2523.167	Ratio 2.1 ≤ (X^2 /d.f) ≤ 3.1 (Cheung & Rensvold, 2002)	χ^2 /d.f is within the range of threshold, shows the model is good
d.f	998		
X^2 /d.f	2.528		
RMSEA	0.058	0.05≤ (RMSEA) ≤ 0.080 (Dion, 2008)	Less than the minimum threshold, shows the model is good
CFI	0.912	≥ 0.950 (Dion, 2008)	Less than the threshold, suggests model needs modifications
GFI	0.809	≥ 0.90 (Cheung & Rensvold, 2002)	Less than the threshold, shows the model needs modification
SRMR	0.0437	SRMR ≤ 0.08 (Dion, 2008)	In acceptable range 0 ≤ SRMR ≤ 0.09, shows, model is good

5.6.2 The model fits improvement

Table 5.11 shows that the fit indices CFI and GFI indicate that model modification is required. As noted by Dion (2008) it is doubtful that one will have all measurements of fit indices showing best fit without model modification. AMOS v 23.0 allows one to set the threshold for refinement indices by identifying the level χ^2 change that must be incorporated in the output with the default being 4.00 or close to 4.00. The AMOS v 23.0 output was recomputed and the following elimination principles recommended by Dion (2008) and Cheung and Rensvold (2002) were used to achieve the best fit.

1. All standardised regression weight values must be above 0.5 (preferably above 0.7).
2. All squared multiple correlations must be above the threshold value of 0.5
3. All standardised residual covariance must be above 2.58 or below -2.58 (known as the absolute value 2.58), and
4. Modification indices that disclose high covariance between measurement errors accompanied by high regression weights between these errors' decision variables are candidates for deletion and must be removed if necessary.

Table 5.12 shows that 18 pairs of residual covariance have the highest modification indices. These either need to be deleted or treated as free parameters. For example, results indicate that if the analysis is repeated when the covariance between e37 and e39 is treated as a free parameter, the discrepancy in the model will fall by at least 85.124. This implies that, the χ^2 statistic of measurement model will drop by 85.124 from the current 1910.941. However, only those variables which showed high covariance as well as high regression weights were deleted. In this model variables with no regression weight were not deleted but were treated as free parameters. The variables that require modification indices and which were treated as free parameters are Org_1, Org_2, Org_3, Org_4, People_3, People_4, Effort_6, Effort_7, Intention_1, Intention_2, Intention_5, Intention_6, Intention_7, Tech_1 and Tech_3, as shown in Table 5.13.

When AMOS v23 was run after treating the modification indices (MI) covariance, all standardised regression weights and all squared multiple correlations were above 0.5, and all standardised residual covariance were above 2.58 or below -2.58.The refined model produced a new set of results which showed a good improvement. The new results described chi-square statistics of 1910.941 with 989 degrees of freedom (χ^2 /d.f = 1910.941/989 = 1.932). The new output shows the values of X^2/d.f which is near to the lower bound of the limits, thus signifying a good fit. In additional, confirmed fit indices produced improved outputs with all results indicating a good fit of the model. RMSEA = 0.045, GFI = 0.900, CFI = 0.950 and SRMR = 0.410 at 90% confidence interval and 0.000 probability level with HI 90.and LO 90. All results suggested a good fit of the model.

Table 5.12: Extract of modification indices from Amos output for covariance (Unrefined)

Error terms			Modification index (M.I.) covariance
e37	<-->	e39	85.124
e35	<-->	e36	35.463
e33	<-->	e34	20.522
e25	<-->	e26	109.187
e21	<-->	e25	28.443
e21	<-->	e22	34.348
e18	<-->	e19	37.913
e16	<-->	e19	31.848
e16	<-->	e17	48.746
e13	<-->	e14	37.753
e12	<-->	e13	35.829
e10	<-->	e11	61.493
e8	<-->	e11	25.992
e8	<-->	e9	64.852
e6	<-->	e7	72.200
e4	<-->	e24	28.694
e4	<-->	e6	22.094
e2	<-->	e7	23.231

Table 5.13: Extract of modification indices from Amos output for regression weights (Unrefined)

Paths			**Modification index Regression weights**
Tech_3	<---	Project_3	21.705
Tech_3	<---	Project_1	20.481
Tech_3	<---	Tech_1	58.996
Tech_1	<---	Tech_3	53.521
Actual_6	<---	Actual_5	27.133
Actual_5	<---	Actual_6	21.383
Actual_4	<---	Effort_4	24.174
Org_3	<---	Effort	29.375
Org_3	<---	Actual_4	20.505
Org_3	<---	Effort_5	21.723
Org_3	<---	Effort_4	31.560
Org_3	<---	Effort_3	25.819
Org_3	<---	Effort_2	31.576
Org_1	<---	Actual_6	20.249
Org_1	<---	Actual_3	21.249
Intention_7	<---	Effort_4	21.148

Table 5.14: Model fit indices with their threshold values (Refined)

Fit indices	**Measurement model**	**Threshold**	**Recommendations for the measurement model**
χ^2	1910.941	Ratio 2.1 ≤ (X^2 /d.f) ≤ 3.1	χ^2 /d.f is within the range of threshold, shows the model is supported (good)
d.f	998		
χ^2 /d.f	1.932		
RMSEA	0.045	0.05≤ (RMSEA) ≤ 0.080	Less than the threshold, shows the model is good
CFI	0.950	≥ 0.950	Equal to the threshold, shows the model is good
GFI	0.900	≥ 0.90	Equal to the threshold, shows the model is good
SRMR	0.0410	SRMR ≤ 0.08	In acceptable range 0 ≤ SRMR ≤ 0.09, shows, model is good

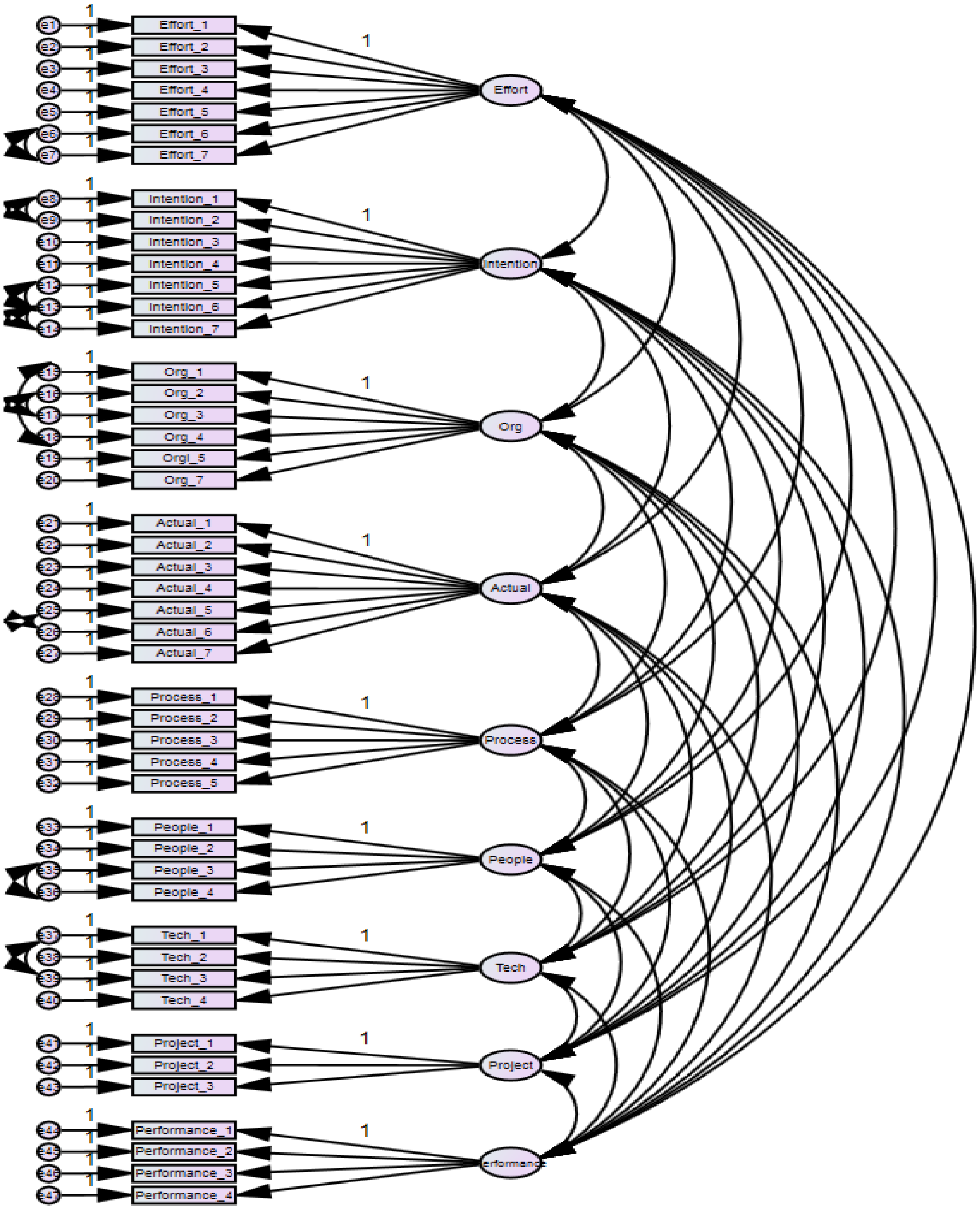


Figure 5.5: Measurement model of critical success factors influencing agile software development projects (Refined)

5.7 The structural model

The study employs structural equation modelling to examine the presented research model**.** Byrne (2013) suggested a two-step model building approach for structural models. The first structural equation model (the measurement structural model) represents the theory specifying how constructs are related to other decision variables (observed) in the model. Its key components consist of latent variables that are measured by several indicators (Byrne, 2001). Every measurement indicator is connected with an error term. The second structural model identifies the relationships among latent variables as suggested by theory (Byrne et al., 2007). The main difference between these two structural models is that the measurement structural model displays the relationships between the variables (both measured and latent (hidden) while the structural model shows only the relationships between the latent (hidden) variables (Pallant, 2013). The importance of developing a structural model is to identify which decision variables are related and to indicate the magnitude and nature of each relationship. Thus structural equation modelling allows models with complexity to be developed and to show their relationships on their path diagrams.

The South Africa data sample was inputted into AMOS v 23.0 to SPSS v 23.0 for the model fit of CFA. Then the data sample was computed and evaluated as with the measurement models. Every step of model fit testing was implemented, as for the measurement model.

5.7.1 Structural model analysis (South African data sample)

Figure 5.6 below illustrates the structural model for the South African data sample (before modification). The structural model is different from the measurement model (figure 5.5) in that the arrows point from one decision variable to another, in line with the theorised relations in the direction as shown in the research model of the study.

The initial run of the model through the analysis software showed χ^2= 175.378 with degrees of freedom d.f = 11 giving a ratio of χ^2 / d.f =175. 378 / 11 = 15.943 at p=0. 000 significant at p ≤ 0.01. Although the ratio χ^2 / d.f was within the acceptable range (as shown in Table 5.17), the probability less than 0.05 suggests that if one rejects the null hypothesis, then the model has a good fit but might give a minimum error to the model of the study.

As discussed above, this measurement needs to be compared with other fit indices before the model is modified (Dion, 2008; Barrett, 2007). The outputs were: GFI = 0.926, CFI = 0.882 and RMSEA = 0.019. The CFI fit index was below the threshold and the RMSEA was above the threshold (as shown in table 5.17). Thus the model required refinement before the hypotheses could be assesses.

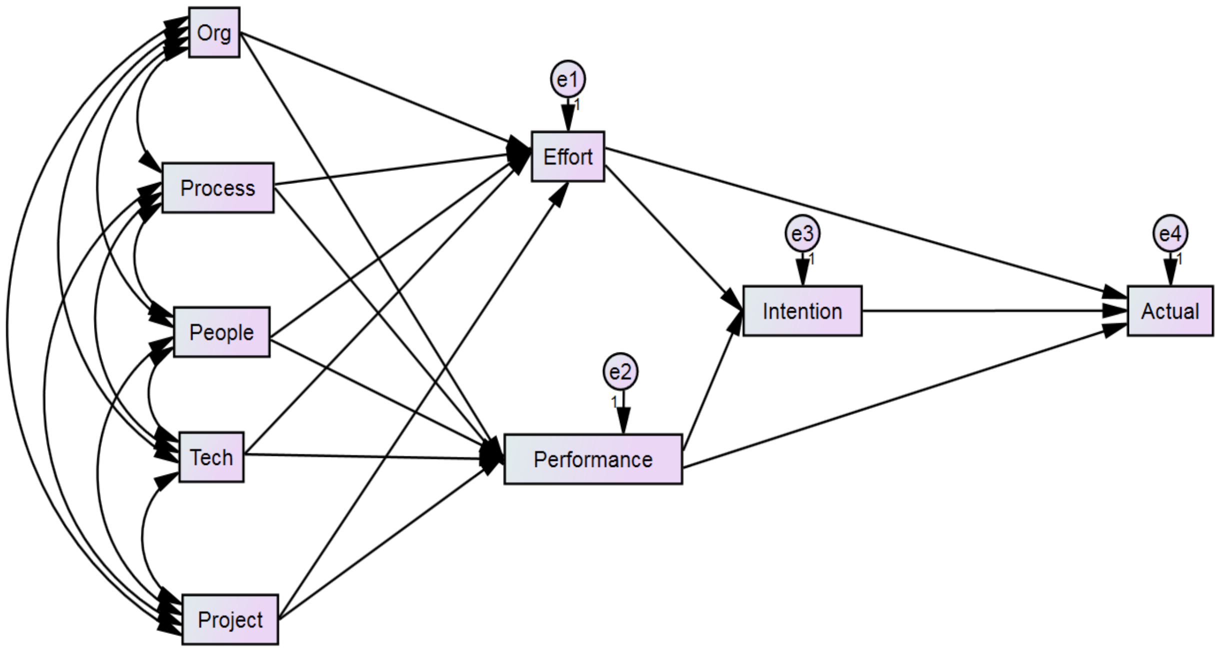


Figure 5.6: Structural model for South Africa data sample (Unrefined)

Note: For each indicator an error term was appended. The error terms were arbitrary names with a term 'e' and a numerical value. To indicate the relationships double headed arrows were used to represent the covariance. The single headed arrows were used to represent the causal relationship between decision variables. Rectangle shapes indicate the constructs.

Byrne and Stewart (2006) recommend, before removing parameters, systematically fixing and checking for potential errors that could cause a structural model not to fit. This includes checking for missing data, non-normality, small sample sizes and multilevel data before any model refinement is implemented. This must be carried out in line with the recommendations made by the modification indices (Dion, 2008). Hence, the matter of the small data sample should not arise. The normality, missing data and multicollinearity issues were evaluated during data screening (as already discussed) and every irregularity resolved. Thereafter, it was vital to evaluate the auto-correlation of parameters, by analysing the regression weights and covariance, as illustrated in Tables 5.15 and 5.16. All modification indices in Table 5.16 were deleted and modification indices for covariance were treated as free parameters by not deleting them (as illustrated in Table 5.15).

Table 5.15: Modification indices for covariance (Unrefined)

Error term			M.I.	Par Change
e1	<-->	e2	73.987	.074
e3	<-->	Tech	4.374	-.017
e3	<-->	Org	4.108	.023
e3	<-->	e2	14.665	-.023
e4	<-->	Process	55.925	.113

Table 5.16: Modification indices for regression weights (Unrefined)

Path			M.I.	Par Change
Performance	<---	Effort	58.210	.140
Effort	<---	Performance	23.721	.291
Intention	<---	Tech	4.179	-.104
Intention	<---	Process	5.934	.069
Intention	<---	Org	7.474	.101
Actual	<---	Process	53.784	.213

Table 5.17: Model fit indices with their threshold values (Unrefined)

Fit indices	Measurement model	Threshold	Recommendations for the measurement model
χ^2	175.378	Ratio 2.1 ≤ (χ^2/d.f) ≤ 3.1	X^2 /d.f is not within the range of threshold, need modification
d.f	11		
χ^2/d.f	15.943		
RMSEA	0.019	0.05≤ (RMSEA) ≤ 0.080	Less than the threshold, shows the model is good
CFI	0.882	≥ 0.950	Less than the threshold, suggests model need modifications
GFI	0.926	≥ 0.90	More than the threshold, shows the model is good
SRMR	0.061	SRMR ≤ 0.08	Acceptable range 0 ≤ SRMR ≤ 0.09 but on the higher side.

Covariance, as illustrated in Table 5.15, was correlated with the displayed error terms.

The modification indices for covariance assist in controlling the value of chi-square (χ^2) thereby decreasing the chi-square (χ^2) value relative to the degree of freedom (d.f). This means as chi-square (χ^2) decreases, the degree of freedom (d.f) must also decrease. This means when three error terms (e1 and e2, and e3) are allowed to converge, their covariance might be anticipated to change by between 0.074 and -0.023, which would reduce the structural model's χ^2 by close to 1.272, leading to a better fit. The process of modification indices for regression weights is better than deletion of parameters of the model (Byrne, 2001).

Post refinement, the covariance of the model was recomputed to check for model fitness. However, it was discovered that not all the recommended refinements made significant changes to the fitness of the model. Positive significant refinements were retained and covariance recomputed.

The refined structural model (as presented in Figure 5.7) was thus computed and the fitness indices were attained. The resulting fitness indices (as shown in Table 5.18) were:

- χ^2 = 10.174 with degrees of freedom d.f = 8 giving a ratio of χ^2 / d.f = 10.174 / 8 = 1.272 with 90% confidence interval at p=0.253 not significant at $p \leq 0.05$,
- GFI = 0.995,
- CFI = 0.998,
- SRMR = 0.0149 and
- RMSEA = 0.024.

Every fit index of the model was within the acceptable range and was significant. Model fitness was thus attained without removing any parameters. There was a higher probability than the threshold of 0.05 values which means that there could be a big error margin that the model does not fit, if the null hypothesis is rejected.

5.7.2 Hypothesis testing of South Africa data

The estimates of the free parameters and the theorised pathways were evaluated according to the acceptable model fit. Figure 5.7 below represents the theorised relationships between the latent variables that form the supporting causal structure of intention to use agile software projects (following Byrne's (2001) and the recommended critical ratio (CR) threshold of above +/-1.96). In addition, the standardised parameter estimates are analysed to come up with the strength of paths, and the explanations of hypotheses within the model as, illustrated in Table 5.19.

Table 5.18: Model fit indices with their threshold values (Refined)

Fit indices	**Measurement model**	**Threshold**	**Recommendations for the measurement model**
χ^2	10.174	Ratio 2.1 ≤ (χ^2 /d.f) ≤ 3.1	χ^2 /d.f is within the range of threshold, shows the model is good
D.f	8		
χ^2 /d.f	1.272		
RMSEA	0.024	0.05≤ (RMSEA) ≤ 0.080	Less than the threshold, shows the model is good
CFI	0.998	≥ 0.950	More than the threshold, shows the model is good
GFI	0.995	≥ 0.90	More than the threshold, shows the model is good
SRMR	0.0149	SRMR ≤ 0.08	Acceptable range 0 ≤ SRMR ≤ 0.09 but on the higher side.

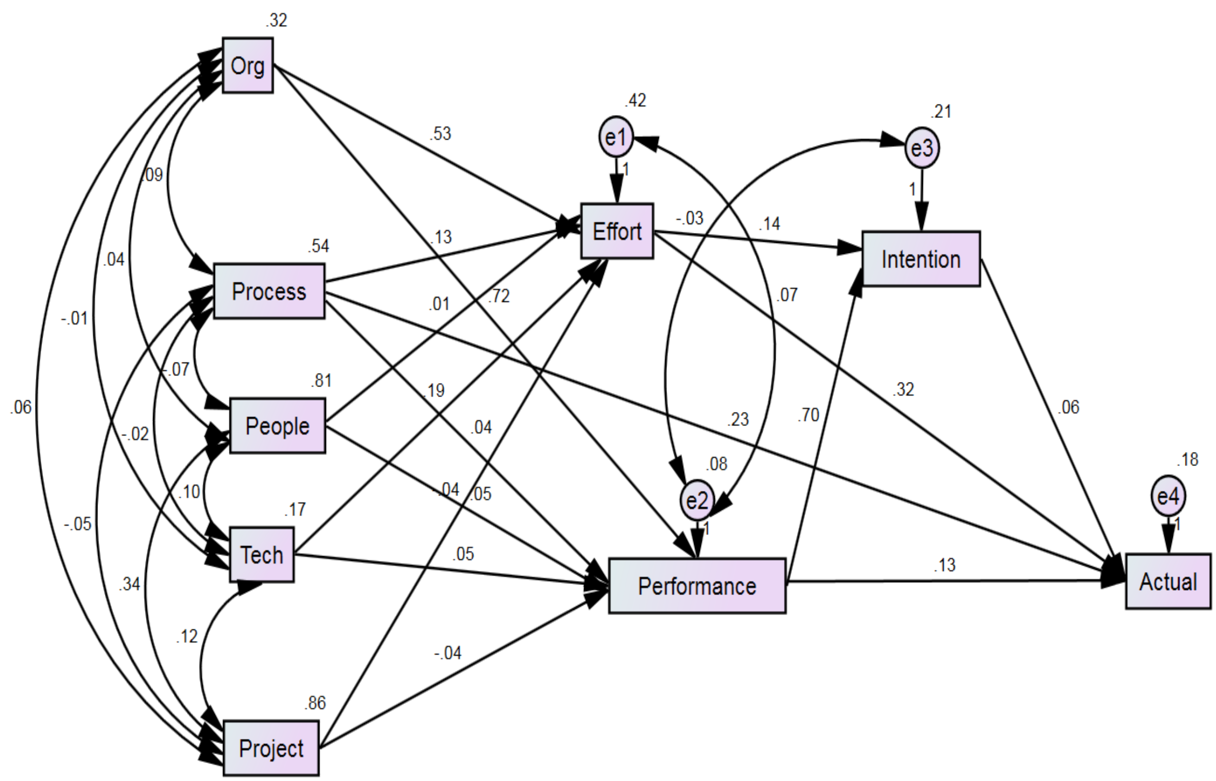


Figure 5.7: Refined structural model for the South Africa data

Table 5.19: Extracted standardised significance levels of the structural model

Hypotheses	Paths			Estimate	S.E.	C.R.	P	Recommendation
H12a	Performance expectancy	<---	Project	-.041	.016	-2.621	.009**	Partially Supported
H11a	Effort expectancy	<---	Process	.132	.043	3.094	.002**	Supported
H10a	Effort expectancy	<---	Organisation	.528	.055	9.547	***	Supported
H11b	Performance expectancy	<---	Process	.036	.018	1.986	.047**	Supported
H8b	Performance expectancy	<---	People	.047	.016	2.926	.003**	Supported
H5c	Performance expectancy	<---	Technological	.047	.034	1.401	.161	Not supported
H12b	Effort expectancy	<---	Project	-.040	.037	-1.083	.279	Not supported
H10b	Performance expectancy	<---	Organisation	.721	.024	29.557	***	Supported
H5b	Effort expectancy	<---	Technological	.185	.078	2.361	.018*	Supported
H8a	Effort expectancy	<---	People	.012	.038	.320	.749	Not supported
H3	Intention behaviour	<---	Effort expectancy	.138	.039	3.545	***	Supported
H2	Intention behaviour	<---	Performance expectancy	.705	.066	10.745	***	Supported
H4	Actual success	<---	Intention	.065	.045	1.454	.146	Not supported
H13a	Actual success	<---	Performance expectancy	.132	.053	2.475	.013*	Supported
H11d	Actual success	<---	Process expectancy	.230	.028	8.185	***	Supported
H3b	Actual success	<---	Effort expectancy	.316	.035	9.113	***	Supported

Note: $\chi 2$ /d.f (10.174/8) = 1.272; *** p < 0.001; ** p < 0.01; * p < 0.05; GFI = 0.995; CFI = 0.998, SRMR = 0.0149; RMSEA = 0.024

Note: p =0.253 which greater than 0.05.

The critical ratio values of the paths between the hypothesised decision variables show that these paths are significant, except those of hypothesis H5c (β=0.047; p> 0.05), H12b (β=-0.040; p> 0.05), H8a (β=0.012; p> 0.05) and H4 (β=0.065; p> 0.05). This means that the proposed hypotheses that technological factors influences performance expectancy factors (H5c), that project factors influence effort expectancy factors (H12b), that people factors influence effort expectancy factors (H8a) and that intention behaviour factors influence actual success factors (H4) are not supported.

Hypothesis H2 (β=0.705; p<0.05) and H3 (β=0.138; p<0.05) indicate the existence of a positive influence between effort expectancy factors and intention behaviour factors, and performance expectancy factors and intention behaviour factors, as shown in Table 5.19. This Table also shows that the following hypothesised relationships for effort expectancy are significant and supported:

- H10a - organisational factors influence effort expectancy factors (β=0.528; p<0.05),
- H11a - process factors influence effort expectancy factors (β=0.132; p<0.05), and
- H5b - technological factors influence effort expectancy factors (β=0.185; p<0.05)

On the same note, the following hypotheses for performance expectancy are significant and supported

- H12a - project factors influence performance expectancy factors (β=-0.041; p<0.05),
- H11b - process factors influence performance expectancy factors (β=0.036; p<0.05),
- H8b - people factors influence performance expectancy factors (β=0.047; p<0.05), and
- H10b - organisational factors influence performance expectancy factors (β=0.721; p<0.05).

Lastly, hypothesis H11d (β=0.230; p<0.05) between process factors and actual success factors is supported and is significant.

Two direct paths to actual success factors which were hypothesised -- H13a (β=0.132; p<0.05) and H3b (β=0.316; p<0.05) for performance expectancy factors and effort expectancy factors – were supported, meaning that they are significant. All other paths were not significant at p< 0.001 meaning that the following hypotheses are not supported in the final model as shown in table 5.19:

- H1 - social influence factors have positive influence towards intention behaviour factors,
- H5a - social influence factors have positive influence towards intention behaviour factors,
- H5c - social influence factors have positive influence intention behaviour factors,

- H11b - social influence factors have positive influence towards intention behaviour factors,
- H6a - cultural factors have positive influence towards actual success factors,
- H6b - cultural factors have positive influence towards intention behaviour factors,
- H6c - cultural factors have positive influence towards actual behaviour success factors,
- H7a - political factors have positive influence towards social influence factors,
- H7b - political factors have positive influence towards effort expectancy factors,
- H8c - people factors have on influence actual success factors,
- H9a - cultural factors have positive influence towards social influence,
- H9b - cultural factors have positive influence towards intention behaviour factors,
- H11c - process factors have positive influence towards intention behaviour factors,
- H12c - project factors have positive influence towards intention factors and H13c - social factors have positive influence towards actual success factors.

The predictions were measured by using the indirect and direct effects of the decision variable as theorised in the model. The extracted standardised coefficient weights output from AMOS v23.0 are displayed in Table 5.20.The standardised coefficient weights represent the direct effects between decision variables. Relying on direct effects, AMOS v 23.0 was computed to produce a matrix table from which the indirect effects of the decision variables may be explained and extracted. Moreover, AMOS v 23.0 computes the direct and the indirect effects to give the total of a decision variable on the dependent variable as shown in Table 5.20. The results of Table 5.20 show that there is a direct effect on success factors among the three decision variables (namely, effort expectancy factors, performance expectancy factors, and process factors) towards the dependent factor.

- Organisational factors have the highest effect towards effort expectancy (H10a; $\beta=0.410$; $p<0.05$);
- Performance expectancy has the highest effect towards intention behaviour factors (H2; $\beta=0.614$; $p<0.05$);
- Organisational factors have the highest effect on performance expectancy factors (H10b, $\beta=0.809$; $p<0.05$); and
- Effort expectancy factors have the highest effect towards actual success factors (H3b, $\beta=0.405$; $p<0.05$)

The framework decision variables explain 18.0% of variance with regards to actual success, 21.0% of variance with regards to intention behaviour factors, 42.0% of variance with regards to effort expectancy factors and 8.0% of variance with regards to performance expectancy factors of agile software development projects.

The total effect of each decision variable on actual success factors can be observed as the effect that that decision variable has towards actual success factors with or without a mediating variable being involved.

The direct effect is the effect a decision variable has on a dependent decision variable, in this case actual success factors with no mediating variable. As per the model above (figure 5.7), performance expectancy, effort expectancy, process and intention behaviour factors where hypothesised to have direct effects to the actual success of agile software projects. Table 5.20 shows that these direct effects exist with values 0.117, 0.405, 0.296 and 0.066 respectively.

Indirect effects are the influences an independent decision variable has on the dependent variable through one or more mediating variables. As per the model, the decision variables effort expectancy and performance expectancy factors are mediated by intention behaviour factors and then influence actual success factors. Thus, the effects of effort and performance expectancy on actual success factors are regarded as indirect effects. The product of the indirect and direct effects form the total effect of an independent decision variable on a dependent one is displayed in Table 5.21.

Table 5.20: Extracted standardised regression weights South Africa sample model

Paths			**Estimate**
Performance expectancy	<---	Project	-.076
Effort expectancy	<---	Process	.133
Effort expectancy	<---	Organisation	.410
Performance expectancy	<---	Process	.053
Performance expectancy	<---	People	.084
Performance expectancy	<---	Technological	.038
Effort expectancy	<---	Project	-.051
Performance expectancy	<---	Organisation	.809
Effort expectancy	<---	Technological	.105
Effort expectancy	<---	People	.015
Intention behaviour	<---	Effort expectancy	.173
Intention behaviour	<---	Performance expectancy	.614
Actual success	<---	Intention behaviour	.066
Actual success	<---	Performance expectancy	.117

Paths			Estimate
Actual success	<---	Process	.296
Actual success	<---	Effort	.405

Table 5.20: Standardised total effects of the South Africa data sample model

	Organisation	**People**	**Process**	**Project**	**Technological**	**Effort expectancy**	**Performance expectancy**	**Intention**
Effort expectancy	.410	.015	.133	-.051	.105	.000	.000	.000
Performance expectancy	.809	.084	.053	-.076	.038	.000	.000	.000
Intention	.567	.054	.055	-.055	.042	.173	.614	.000
Actual	.299	.019	.360	-.033	.050	.417	.158	.066

Table 5.21, indicates the standardised total effects of the decision variable which have values of 0.000 between several constructs. In contrast, the influence of the independent decision variable that does not have a direct effect on the dependent decision variable is achieved as a product of the direct and indirect effects of the two decision variables. For example hypothesis (H11a) projected a positive effect for process factors on actual success when mediated by effort expectancy factors. The effect of process factors on effort expectancy factors is (0.133) which is not significant and that of effort expectancy factors to actual success factors is (0.417) which is also not significant. However the product of the direct and indirect effect of process factors to actual success factors is (0.133) (0.417) which is equal to (0.055). This reasoning thus suggests that hypothesis (H11a) is significant at $p<0.10$, meaning that it is supported. Results indicate that organisational factors have one of the largest total effects on performance expectancy factors (0.809), and that performance expectancy factors have the second highest effect on intentional factors (0.614).

Table 5.21: Original model and final model

Hypotheses	Final Findings	Original Findings
H1: Social influences have a positive effect on intention of the agile methodologies for the success of an agile software development projects.	Not supported	Supported
H2: Performance expectancy has a positive effect on agile intention for the success of an agile software development projects.	Supported	Supported
H3: Effort expectancy has a positive effect on agile intention for the success of the agile software development projects.	Supported	Supported
H4: Behavioural intention has a positive effect on the success factors of an agile software development projects.	Not Supported	Supported
H5a: Technological factors have a positive effect on behavioural intention for the success of agile software development projects.	Not Supported	Supported
H5b: Technological factors have a positive effect on effort expectancy for the success of the agile software development projects.	Supported	Supported
H5c: Technological factors have a positive effect on performance expectations for the success of agile software development projects.	Not Supported	Supported
H5d: Technological factors have a positive effect on the success of the agile software development projects	Not Supported	Supported
H11a: Process factors have a positive effect on effort expectancy for the success of the agile software development projects.	Supported	Supported
H11b: Process factors have a positive effect on performance expectations for the success of the agile software development projects.	Supported	Supported
H11c: Process factors have a positive effect on behavioural intention for the success of the agile software development projects.	Not Supported	Supported
H11d: Process factors have a positive effect on the success of the agile software development projects.	Supported	Supported
H12a: Project factors have a positive effect on performance expectations for the success of the agile software development projects.	Partially Supported	Supported

Hypotheses	Final Findings	Original Findings
H12b: Project factors have a positive effect on effort expectancy for the success of the agile software development projects.	Not Supported	Supported
H12c: Project factors have a positive effect on behavioural intention for the success of the agile software development projects.	Not Supported	Supported
H12d: Project factors have a positive effect on the success of the agile software development projects.	Not Supported	Supported
H13a: Performance expectancy has a positive effect on the success of the agile software development projects.	Supported	Supported
H13b: Effort expectancy has a positive effect on the success of the agile software development projects.	Supported	Supported
H13c: Social Influence has a positive effect on the success of the agile software development projects.	Not Supported	Supported
H6a: Culture have a positive effect on the success of the agile software development projects.	Not Supported	Supported
H6b: Political have a positive effect on the behavioural intention success of the agile software development projects.	Not Supported	Supported
H6c: Political factors have a positive effect on the organisational success factors of the agile software development projects.	Not Supported	Supported
H7a: Political factors have a positive effect on agile professionals' social influence for the success of the agile software development projects.	Not Supported	Supported
H7b: Political factors have a positive effect on agile professionals' effort expectancy for the success of the agile software development projects.	Not Supported	Supported
H8a: People factors have a positive effect on effort expectancy factors for the success of the agile software development projects.	Not Supported	Supported
H8b: People factors have a positive influence on performance expectancy factors for the of the agile software development projects.	Supported	Supported

Hypotheses	Final Findings	Original Findings
H8c: People factors have a positive influence on social influence factors for the success of the agile software development projects.	Not Supported	Supported
H8d: People factors have a positive effect on the overall perceived success of the agile software development projects.	Not Supported	Supported
H9a: Culture has a positive effect on social influences for the success of the agile software development projects.	Not Supported	Supported
H9b: Culture has a positive effect on behavioural intention for the success of the agile software development projects.	Not Supported	Supported
H10a: Organisational factors have a positive effect on effort expectancy for the success of the agile software development projects.	Supported	Supported
H10b: Organisational factors have a positive effect on performance expectancy for the success of the agile software development projects.	Supported	Supported
H10c: Organisational factors have a positive effect on the success of the agile software development projects.	Not Supported	Supported
H14: Gender moderates the relationships among the proposed model constructs	Not Supported	Supported
H15: Age moderate the relationships among the proposed model constructs	Not Supported	Supported
H16: Experience moderates the relationships among the proposed model constructs	Not Supported	Supported
H17: Educational level moderates the relationships among the proposed model constructs	Supported	Supported
H14a: Gender positively moderates the relationships between individual performance expectancy factors and overall success of agile software development projects.	Not supported	Supported
H14b: Gender positively moderates the relationships between individual effort expectancy factors and overall success of agile software development projects.	Supported	Supported
H14c: Gender positively moderates the relationships between project factors and performance expectancy factors for the success of agile software development projects.	Not supported	Supported

Hypotheses	Final Findings	Original Findings
H14d: Gender positively moderates the relationships between people factors and performance expectancy factors for the success of agile software development projects.	Not supported	Supported
H15a: Age positively moderates the relationships between process factors and performance expectancy factors for the success of agile software development projects.	Not supported	Supported
H15b: Age positively moderates the relationships between performance expectancy factors and intention factors for the success of agile software development projects.	Supported	Supported
H15c: Age positively moderates the relationships between effort expectancy factors and intention factors for the success of agile software development projects.	Supported	Supported
H16a: Experience positively moderated the relationships between effort expectancy factors and success factors of the agile software development projects.	Supported	Supported
H16b: Experience positively moderated the relationships between process factors and performance expectancy factors of the agile software development projects.	Not supported	Supported
H16c: Experience positively moderated the relationships between process factors and effort expectancy factors of the agile software development projects.	Not supported	Supported
H17a: Educational level positively moderates the relationships between effort expectancy factors and success of agile software development projects.	Supported	Supported
H17b: Educational level positively moderates the relationships between process factors and performance expectancy factors for the success of agile software development projects.	Not supported	Supported
H17c: Educational level positively moderates the relationships between process factors and effort expectancy factors for the success of agile software development projects.	Not supported	Supported

5.8 Summary

This chapter presented the demographics of the respondents that included age, gender, level of education and industrial experience with regards to agile software projects. The researcher dealt with the raw data that was acquired from the research questionnaires. The data were screened to observe for omissions, standardised deviation, kurtosis, skewness and to correct the discrepancies that might have occurred during its capture. The Mahalanobis distance method was implemented to identify and fix univariate and multivariate outliers. In addition, checking for the existence of multicollinearity and non-normality was implemented and a probability-probability (p-p) plot was used to show the findings. The researcher discussed the validation and analysis of the research model using Cronbach alpha, Confirmatory Factor Analysis (CFA) and Exploratory Factor Analysis (EFA). The measurement model was presented using Structure Equation Modelling (SEM) and confirmatory factor analysis. A sequential process of creating the measurement fit-indices were carried out to support the appropriateness of the model for the South African data. Some parameters of the decision variables were out of scope and had to be adjusted in order to come up with a more refined and better structural equation model. The final model was implemented to test the hypotheses that had been formulated in the literature review chapter of the study. The framework decision variables explain 18.0% of variance with regards to actual success factors, 21.0% of variance with regards to intention factors, 42.0% of variance with regards to effort expectancy factors and 8.0% of variance with regards to performance expectancy of agile software development project. Based on the results of the structural model, a generic model was established that was supported and confirmed as the closest fit for the data sample of South Africa.

The next chapter discusses the results that were obtained in the multi-group analysis to prove the set of hypotheses.

CHAPTER 6: THE EFFECT OF MODERATORS

6.1 Introduction

This chapter explains the effect of moderators namely: gender, age, level of education and experience of work on agile software projects. The influence of the moderators on the decision variables of the research model is examined through use of measurement invariance and multiple group analysis invariance (Byrne & Stewart, 2006).

Measurement invariance refers to the degree to which items or components of the construct have equal meanings across the groups which are examined, by looking at two levels of the research (Byrne et al., 2007). The first level examines the equivalence of the psychometric properties of the instrument, such as measurement, configural and metric errors (Byrne, 2001; Byrne, 2013). The second level examines the group differences using covariance and latent means analysis (Barrett, 2007; Byrne, 2001). Multiple group, confirmatory factor analysis (MGCFA) is an error measurement method which mainly focuses on investigating levels of measurement invariance, thereby allowing invariance tests to be carried out across the two groups simultaneously (Pomykalski et al., 2008).

In addition, when investigating for invariance, the fit of the configural invariance provides the standard value against which all successively specified invariance models are compared (Schreiber, 2008). Configural invariance refers to the initial step in the process which needs the number of indictors and factor loading pattern to be the same across groups, although the factor loadings themselves might differ among the groups (Byrne et al., 2007). This means that the same parameters that were individually approximated in the baseline model for each group are again approximated in the multiple group model. Configural invariance uses factor loading, residual and intercept invariance as the most common methods for first order factor models (Rohani et al., 2009).

The factor loadings invariance (metric invariance) approach focuses on the equality of constraints to be quantified for all freely estimated first-order factor loadings (Byrne, 2013). The next approach after factor loading is factorial invariance. Factorial invariance relies on the comparison of the causal effect between the factors and external variables, particularly across groups (Cheung & Rensvold, 2002). However, the configural level of invariance does not support the scales of factors that have common intercepts, thus the need for a second level of invariance examination of the mean, as well as intercept invariance analysis (Dion, 2008).

During intercept invariance testing equality constraints are placed on both the first-order factor loadings and on the observed variable intercepts (Cheung & Rensvold, 2002; Fabrigar et al., 2010). Intercept invariance investigation assists to determine whether any difference between groups is a measurement artefact or a true group difference (Golob, 2003). This research used the following invariance analysis process (Byrne & Stewart, 2006; Byrne et al., 2007):

1. Multiple group covariance structure analysis was used to assess measurement invariance of the four levels of invariant output, namely: structural weights, measurement residuals, measurement weights and structural residuals (Chen et al., 2005). The research focused on the measurement weights and then structural weights, thereby examining the scalar invariance using mean and covariance structure analysis to evaluate the differences in mean scores and to compare means between the two groups.
2. If the measurement weights (metric invariance) were non-invariant, the variant factor loadings were used to constrain the factors equally, one at a time. Thus allowing the measurement weight invariance to have solutions on the item-variable presentation in each empirical conceptual framework and item loading on each causal effect variable (French & Finch, 2006). Thus, if measurement weights are not significantly different across the groups under examination, it is anticipated that the metric is rationally non-invariant (Byrne, 2013).
3. If the structural weight level of invariance was not significant or supported, the causal effect paths of the model were considered non-equivalent between the groups and the paths were constrained one at a time to find the significant or insignificant path differences (Lacobucci, 2010). This was done by means of the significance of changes in chi-square. Thus, if the constrained path produced a significant chi-square change with a p value less than 0.05, it meant that the two groups did not have factorial invariance with respect to that path of causal effect (Byrne, 2001).

6.2 Procedures for determining the effects of moderating factors

Using the mean and covariance structure analysis (MACS) model requires adding constraints to the mean (Chen et al., 2005). The first-order structure includes the following phases: (i) all the factor loadings are constrained equal between the two groups except those fixed to one in the research model, (ii) one group is regarded as a reference group, and the latent means are constrained to zero, (iii) the latent mean of the other group is compared and approximated to the reference group, and (iv) all the factor intercepts are constrained equal between the two groups (Byrne & Stewart, 2006; Byrne et al., 2007).

In the higher order structure, constraints are enforced as follows, if necessary:

i. All the first order factor loadings, apart from those with a fixed value of one, are included, and intercepts are constrained equal between groups as well.
ii. All the higher order factor loadings are constrained equal between the groups, all the higher order intercepts are constrained equal between groups, and the higher order factor mean is approximated for one group and fixed to zero for the reference group (Byrne & Stewart, 2006; Chen et al., 2005).

In the higher order structure, under-identification is a common challenge in assessing for latent mean intercept with regards to the intercepts of the first order, as the number of approximated intercepts surpasses the number of observed measures in a single group (Byrne & Stewart, 2006). Therefore, the under-identification challenge must be dealt with before invariance evaluation of the higher-order model intercept is undertaken. Byrne and Stewart (2006) explained the issue of under-identification and described three model specifications that can be used in evaluating latent mean differences associated with the causal effect of higher order factor structures. The strategy involved in the research models is to constrain the first order latent factor intercept to zero (Byrne, 2013).

Accordingly, Byrne and Stewart (2006) suggest that when using the multiple group analysis, the first-order latent means should be constrained to zero for both groups, which is the same as constraining the four latent factors' intercepts equal across groups. Thus, the number of approximated intercepts is abridged from 16 to 12 for the South Africa sample, thereby, providing a just acknowledged structure that is verified quantitatively. In this scenario, the higher-order factor mean is appropriate to justify for mean differences across actual success factors of the agile software project whilst theorising the four latent means to be approximately equal in magnitude.

The *z* test value associated with the estimates for the South African group serves as the test for significance of the latent factor mean differences between the two groups (Byrne et al., 2007). If the mean approximated values of the non-reference group are positive, it is explained that this group has higher means than the reference group, and if mean approximated values were negative, the reference groups is explained to have higher means.

6.3 The South African sample

The demographic information for age, level of education, and experience (but excluding gender) was recorded through use of the median split method in SPSS v 23.0 to enable computation of the group analysis for investigation. Figures 6.1 and 6.2 illustrate the unconstrained structural model and a multiple group analyses which were conducted with regards to four demographic variables of the South African data sample. The results are discussed for each demographic moderators separately below.

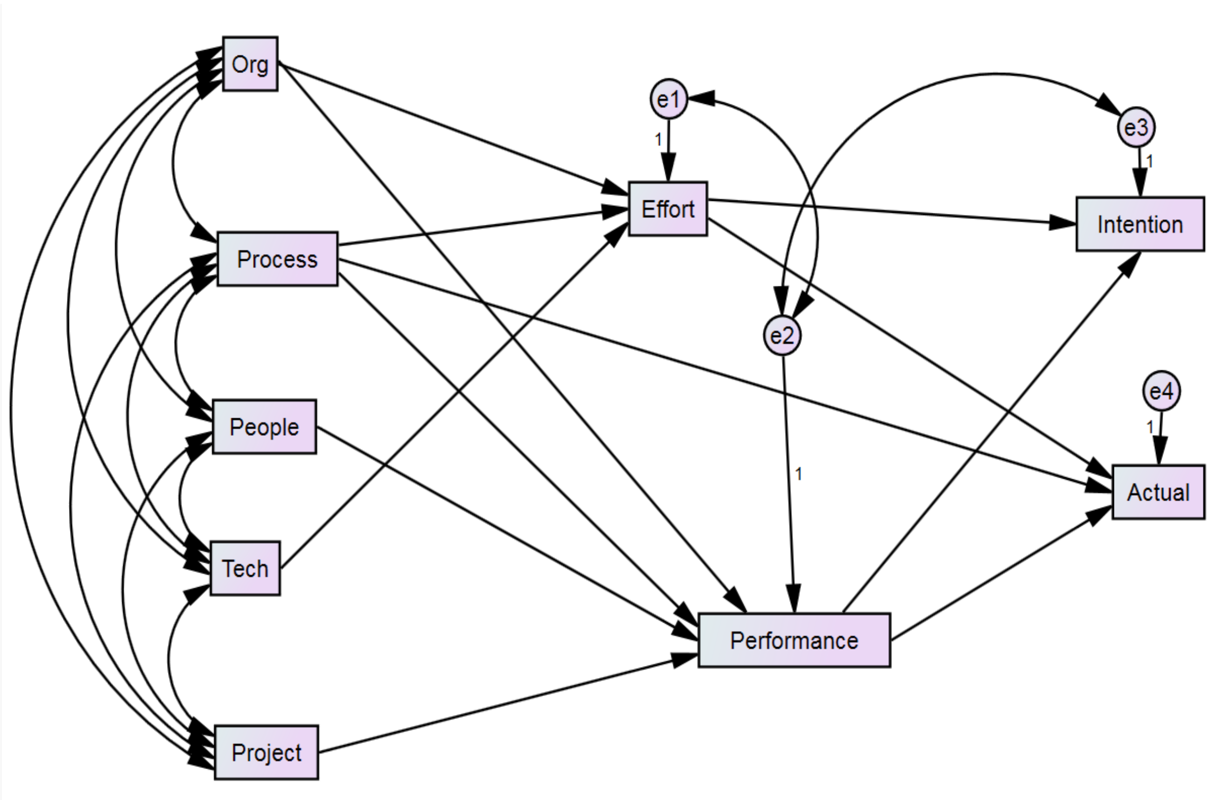


Figure 6.1: The South Africa covariance structure model (Unconstrained structural model)

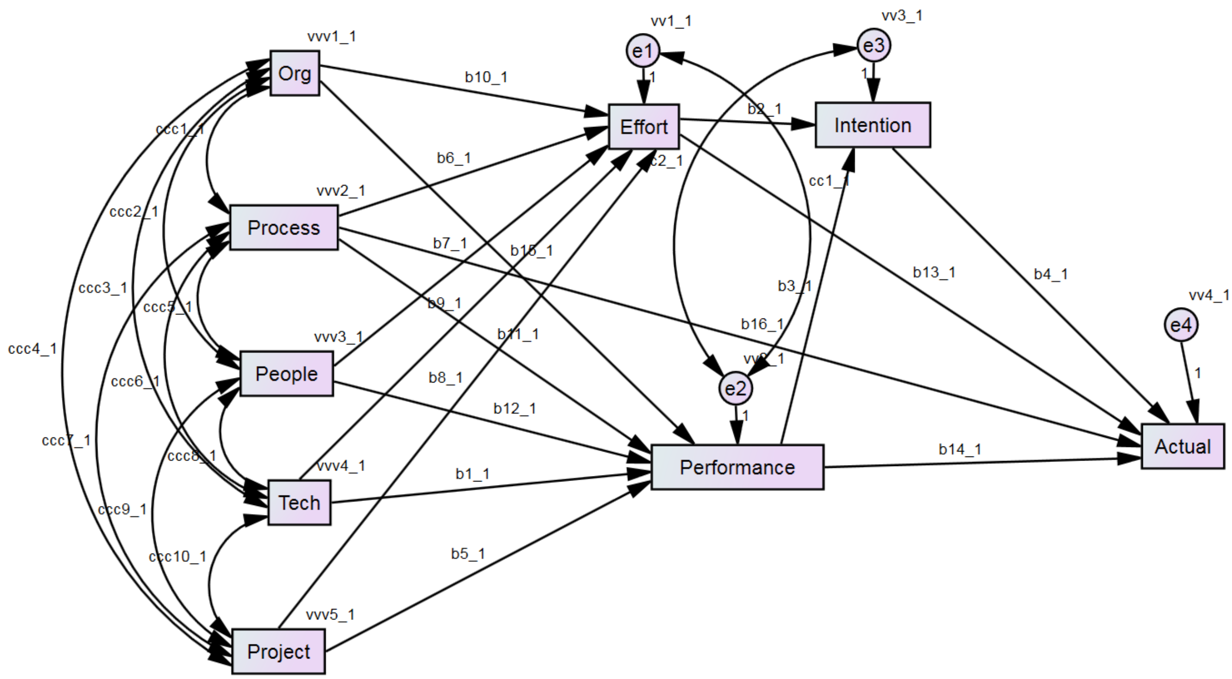


Figure 6.2: Constrained structural model (Closest invariant model of critical success factors for agile software development projects used in the South African data sample)

6.3.1 Gender influence

The South Africa sample consisted of 297 males and 163 females. That means, there are approximately twice many males as females. Following the guidelines by Byrne (2001) and Cheung and Rensvold (2002) the final structural model depicted in Figure 6.2 is applied first to males and females separately to examine if each group can achieve an adequate fit separately (Hair et al., 2006). The final structural model shown in Figure 6.1 was used to discover if each group (males and females) can accomplish an acceptable fit (Dion, 2008).

The males' sample fit statistical results are:

- CMIN = 3.553 with d.f =6 and CMIN/d.f ratio = 0.592;
- SRMR = 0.015;
- CFI =1.000; and
- RMSEA =0.000 with 90 percent confidence interval (0.000 and 0.066) and PCLOSE =0.808.

The females' sample fit statistics are:

- CMIN = 9.833 with d.f =7 and CMIN/d.f ratio = 1.405;
- SRMR = 0.036;
- CFI = 0.994; and
- RMSEA = 0.050 with 90 percent confidence interval (0.000, 0.116) and PCLOSE = 0.435

These results all indicate a good fit and one can conclude that the study established an acceptable model fit for the gender sample. A group multiple analysis test for factor structure equivalence was computed for the two groups concurrently. The estimation results and critical ratio are shown in Table 6.1 below.

All paths are significant for both gender groups, except:

- Performance (Performance expectancy factors) - Project (Project factors) for males,
- Effort (Effort expectancy factors) - Tech (Technological factors) for males,
- Performance (Performance expectancy factors) - Process (Process factors) for males,
- Performance (Performance expectancy factors) - People (People factors) for females, and
- Actual (Actual success) - Performance (Performance expectancy factors) for females.

The AMOS model comparison results revealed that the two groups (female and male) are invariant. Supposing the unconstrained model to be correct, the measurement weights between the two gender groups is invariant as is shown by the non-significant p values which are greater than 0.05 in Table 6.2 (Byrne, 2001).

Also the male and female groups are invariant on the structural weights, assuming the measurement model to be correct.

In the study it was assumed that the factor loadings are equal between the two groups. Therefore, assuming as well that the structure weights are correct, the structural residuals are invariant between the two groups, depicted by the non-significant p value results.

Most researchers stop at the structural weights level of investigation and assume that examining residual invariant is more stringent than necessary (Byrne & Stewart, 2006; Byrne et al., 2007). Accordingly, the residual invariance is not tested in the following sections.

Table 6.1: South Africa's un-standardised estimates and critical ratios for gender groups

Paths			Male		Female		Notes
			Estimate	C.R.	Estimate	C.R.	
Performance	<---	Project	-0.027	-1.459 n/s	-0.041	-2.032*	The path is not sig. for males
Effort	<---	Process	0.12	2.265*	0.159	2.223*	
Effort	<---	Tech	0.057	0.676 n/s	0.241	2.079*	The path is not sig. for males
Effort	<---	Org	0.504	7.742***	0.563	5.504***	
Performance	<---	Process	0.028	1.165 n/s	0.052	1.969*	The path is not sig. for males
Performance	<---	People	0.064	3.382***	0.017	0.807 n/s	The path is not sig. for females
Performance	<---	Org	0.697	22.601***	0.756	19.667***	
Intention	<---	Effort	0.127	2.501*	0.146	2.479*	
Intention	<---	Performance	0.783	9.412***	0.568	5.44***	
Actual	<---	Effort	0.291	7.143***	0.391	6.875***	
Actual	<---	Performance	0.201	3.557***	0.107	1.186 n/s	The path is not sig. for females
Actual	<---	Process	0.25	7.378***	0.206	4.112***	
Note: *** p<.001; ** p<.01; * p<.05							

Table 6.2: Multiple group covariance structure analysis for gender groups

Model	DF	CMIN	P	NFI Delta-1	IFI Delta-2	RFI rho-1	TLI rho2
Measurement weights	21	26.680	.182	.018	.019	.002	.002
Structural weights	12	14.863	.249	.010	.010	.002	.003

As shown in Table 6.1 the following hypothesis was supported on both female and male:

H14b: Gender positively moderates the relationships between individual effort expectancy factors and overall success of agile software development projects.

As shown in Table 6.1 as well the following hypothesis were not supported on both female and male:

H14a: Gender positively moderates the relationships between individual performance expectancy factors and overall success of agile software development projects.

H14c: Gender positively moderates the relationships between project factors and performance expectancy factors for the success of agile software development projects.

H14d: Gender positively moderates the relationships between people factors and performance expectancy factors for the success of agile software development projects.

Evidence of invariance has been established based on the differences in the chi-square test. If the value of change or difference in chi-square is non-significant, it is recommended that the constraints identified in the additional restricted model remain true (Byrne & Stewart, 2006; Byrne et al., 2007). Researchers have disputed that the difference or change in chi-square and chi-square value is sensitive to sample size and non-normality, thus rendering it an unfeasible and impractical standard on which to base evidence of invariance or equality (Dion, 2008). Thus, there has been a tendency to dispute evidence of invariance based on change in the CFI value between models as being non-significantly small (Pomykalski et al., 2008). The differences in CFI must not exceed 0.01 as stipulated by different researchers (Byrne & Stewart, 2006; Golob, 2003; Iacobucci, 2010; McIntosh, 2007). The model fit results for the multiple group analysis discovered acceptable an output with respect to changes in CFI of less than 0.01 (as depicted in Table 6.2). Consequently, it is has been concluded that the model is functioning homogeneously across males and females.

Table 6.3: Multiple group covariance structure analysis for gender groups

Model	**NFI Delta1**	**RFI rho1**	**IFI Delta2**	**TLI rho2**	**CFI**	**ΔCFI**
Unconstrained	.982	.947	.999	.995	.998	
Measurement weights	.980	.947	.999	.995	.998	0.000

Structural weights	.972	.944	.996	.993	.996	0.002
Structural co-variances	.963	.947	.997	.996	.997	0.001
Structural residuals	.954	.942	.992	.990	.992	0.005
Saturated model	1.000		1.000		1.000	
Independence model	.000	.000	.000	.000	.000	

An important consequence of establishing measurement weights invariance is the possibility of carrying out the subsequent tests of invariance, differences in structural weights and latent means (Barrett, 2007). The level of examination includes an approximation of means based on latent and observed variables through use of the co-variance and mean structure analysis (Pomykalski et al., 2008; Rohani et al., 2009). Using the recommendations stated in subdivision 6.2 and AMOS graphics v 23.0, while constraining the female group to be the reference group, the scale invariance indicates that the male group's mean scores are positive, meaning that the male group has higher mean scores compared to the female reference group.

Table 6.4: Scalar means estimates for South African gender groups

Factors	**Estimate**	**S.E.**	**C.R.**	**P**
Org	3.511	.034	101.974	***
Process	2.091	.042	49.431	***
People	3.212	.052	61.533	***
Tech	.899	.024	38.140	***
Project	3.091	.054	57.616	***

The results show that all the mean scores for the main variables are higher for males than they are for females and the differences are significant since the p values are less than 0.05. Based on the p values and critical ratios, as indicated in Table 6.4, the non-reference group mean scores are statistically different from the reference group in respect to Org, Process, People, Tech and Project.

Byrne (2013) recommended the use of multiple fit indices for evaluating the model goodness of fit in explaining the outputs of mean and covariance structure, which comprise SRMR, CFI, and RMSEA. The model fit statistical outputs for these indices are CFI=26.162, SRMR=0.021 and RMSEA=0.014 with a 90 percent confidence interval (0.000 and 0.041) and PCLOSE=0.991, all indicating a good fit.

Lastly, the invariant output at the structural weight level indicates that factor loadings for the structural paths are not significantly different between females and males. Hence, the hypothesised moderating influence of gender H14 is not supported for the South African sample.

6.3.2 Age influence

The age variable was categorised into five groups. The respondents' demographic attributes indicate that 54.8% are 30 years and below and 45.2% are above 30 years. Thus the age variable was divided into two groups that of above 30 years (regarded as group B) and that of 30 years and below (regarded as group A).

Computing the model for each group independently resulted in values for age group A which all indicate an acceptable fit:

- CMIN= 1.621 with d.f = 2 and CMIN/d.f ratio=0.811;
- SRMR= .0116;
- CFI =1.000; and
- RMSEA = 0.000 with 90 percent confidence interval (0.000 and 0.118) and PCLOSE = 0.633.

The model fit values for the second age group B are as follows:

- CMIN= 5.464 with d.f =4 and CMIN/d.f ratio = 1.366;
- SRMR = 0.026;
- CFI = 0.997; and
- RMSEA = 0.042 with 90 percent confidence interval (0.000 and 0.0.120) and PCLOSE = 0.476.

The model fit is thus within acceptable ranges (supported).

By computing the multiple group analysis the result of the model's estimates and critical ratios are shown in Table 6.5 below.

All paths are significant for age group A except the path linking Actual (Actual Success) and Performance (Performance expectancy factors), while two paths are non-significant for age group B, namely the paths linking Effort - Process and Performance - Process.

Computing the multiple group analysis to test for structural model fit equivalence of the two groups concurrently led to equivalent measurement invariance. Accepting the assumption that the unconstrained model is correct, both groups' measurement weights are invariant as evident from the non-significant p value which is greater than 0.05.

By accepting the assumption that the model measurement weights are correct, it can be concluded that the structural weights are not equal or invariant as the p value is greater than 0.05.

The measurement invariance outputs (as shown in Table 6.6) indicate the difference or the changes in CFI, which do not exceed the 0.01, hence supporting the measurement invariance outputs.

The latent means score differences between the two age groups was checked after establishing measurement invariance. Computing the mean and covariance structure analysis, at the same time constraining age group B to be the reference group, the scalar mean scores indicates that the younger group (age 30 and below) has lower mean scores than the older group (age 30 and above).

Additionally, the two age groups statistically differ on mean scores of latent variables Effort and Performance, although they do not differ significantly on Intention and Actual at $p>0.01$.

Table 6.5: South Africa's un-standardised estimates and critical ratio for age level groups

Paths			Age group A		Age group B		Notes
			Estimate	C.R.	Estimate	C.R.	
Effort	<---	Process	0.132	2.485*	0.128	1.849 n/s	Path not sig. for age group B
Effort	<---	Org	0.546	8.508***	0.478	4.874 ***	
Performance	<---	Process	0.063	2.698**	-0.002	-0.068 n/s	Path not sig. for age group B
Performance	<---	Org	0.728	24.624***	0.711	16.99 ***	
Intention	<---	Effort	0.122	2.276*	0.144	2.497 *	
Intention	<---	Performance	0.69	8.395***	0.752	6.819 ***	
Actual	<---	Performance	0.125	1.902 n/s	0.213	2.962 **	Path not sig. for age group A
Actual	<---	Process	0.243	6.12***	0.224	5.692 ***	
Actual	<---	Effort	0.368	7.492***	0.289	6.436 ***	
Note: *** p<. 001; ** p<. 01; * p<. 05							

Table 6.6: Multiple group covariance structure analysis for age groups

Model	DF	CMIN		P	NFI Delta-1	IFI Delta-2	RFI rho-1	TLI rho2
Structural weights	9	6.729		.665	.005	.005	-.008	-.008
Measurement weights	12	13.032		.367	.010	.010	-.004	-.004

Table 6.7: Baseline comparisons for South African age groups

Model	**NFI Delta1**	**RFI rho1**	**IFI Delta2**	**TLI rho2**	**CFI**	**ΧCFI**
Unconstrained	.994	.969	.998	.992	.998	
Measurement weights	.993	.965	.997	.991	.997	.001
Structural weights	.988	.977	1.000	1.001	1.000	.003
Structural co-variances	.983	.972	.998	.996	.998	.002
Structural residuals	.974	.967	.993	.991	.993	.005
Saturated model	1.000		1.000		1.000	
Independence model	.000	.000	.000	.000	.000	

Table 6.8: Scalar means estimates for South African age groups

Path	**Estimate**	**S.E.**	**C.R.**	**P**
Effort	1.083	.237	4.568	***
Performance	.974	.109	8.918	***
Intention	.555	.230	2.419	.016
Actual	.114	.202	.566	.571

As shown in Table 6.5 the following hypothesis were supported on both female and male:

H15b: Age positively moderates the relationships between performance expectancy factors and intention factors for the success of agile software development projects.
H15c: Age positively moderates the relationships between effort expectancy factors and intention factors for the success of agile software development projects.

As shown in Table 6.5 the following hypothesis was not supported on both female and male:

H15a: Age positively moderates the relationships between process factors and performance expectancy factors for the success of agile software development projects.

Overall model fit values indicate a good model fit:

- CMIN = 7.922 with d.f = 6 and CMIN/d.f = 1.320;

- SRMR = 0.018; CFI =0. 998; and
- RMSEA = 0.026 with 90 percent confidence interval (0.000 and 0.070) and PCLOSE =0. 771

The result of invariance at the structural weight level indicates that the structural paths are not significantly different between the two age groups. Thus, the hypothesised age moderating effect H15 is not supported for the South African sample.

6.3.3 Education influence

The level of education variable was divided into five groups. Combined, the certificate and diploma groups constituted about 49.3% of the sample. The PhD and Master of Science, honours degree and Bachelor of Technology degree, and Bachelor Degree level groups contributed 2.8%, 5.9% and 42.0% of the sample respectively. Thus the education level variable was divided into two groups that of certificate and diploma or below (regarded as group A) and that of a bachelor degree and above (regarded as group B).

Computing the model for each group independently resulted in an acceptable model fit for education group A:

- CMIN= 5.459 with d.f = 7 and CMIN/D.F ratio=0.799;
- SRMR= 0.006;
- CFI =1.000; and
- RMSEA = 0.000 with 90 percent confidence interval (0.000 and 0.008) and PCLOSE = 0.569.

The model fit values for the education group B are also within acceptable ranges (supported):

- CMIN= 11.348 with d.f =8 and CMIN/d.f ratio = 1.419;
- SRMR = 0.026;
- CFI = 0.997; and
- RMSEA = 0.042 with 90 percent confidence interval (0.000 and 0.090) and PCLOSE = 0.476.

The study thus established an acceptable model fit for the education samples. A group multiple analysis test for factor structure equivalence was computed for the two groups concurrently. The estimation results and critical ratio are shown in Table 6.9 below.

All paths are significant for both educational groups, except the following (as shown in Table 6.9):

- Group A:
  - Effort – Process,
  - Performance – Process,
  - Actual – Performance, and
- Group B:
  - Performance – Project,
  - Effort – Tech,
  - Intention – Effort.

The AMOS v 23.0 results of the model comparison revealed that the two age groups are invariant. Supposing the unconstrained model to be correct, the measurement weights between the two educational groups is invariant as shown by the non-significant p value which is greater than 0.05 (shown in Table 6.10).

By accepting the assuming that the model measurement weights are correct, it is possible to conclude that the structural weights are equal or non-invariant, and the p value is less than 0.05.

The measurement invariance outputs (as shown in Table 6.10) indicate the difference or changes in CFI, which do not exceed 0.01, thus supporting the measurement invariance outputs. The structural weights outputs (as shown in Table 6.10) indicate the changes in CFI, which do exceed the 0.01; hence not supporting the structural invariance outputs.

The latent means score differences between the two educational groups were checked after establishing measurement invariance. Computing the mean and covariance structure analysis, at the same time constraining educational group B to be the reference group, the scalar mean scores indicates that the education level below diploma has a negative effect, showing that the reference group has higher mean scores than the diploma and below group (as shown in Table 6.9).

The model fit indices indicate a good fit:

- CMIN = 11.415 with d.f = 14 and CMIN/D.F ratio = 0.815;
- standardised RMR = 0.015;
- CFI = 1.000; and
- RMSEA = 0.000 with 90 per cent confidence interval (0.000 and 0.195) and PCLOSE =0.992.

The structural weights have been shown to be non-invariant. In order to locate the non-equal paths, Byrne and Stewart (2006) outline a procedure that calls for unlabelled structural path coefficients (meaning they are no longer constrained to be equal) and then retests for invariance between the two groups The invariant paths were evaluated one path at a time and the non-equality between the groups was discovered. The model invariance between the two groups is acknowledged through the use of model fit measurements. Hence, measurement invariance was computed with one path constrained at a time. Table 6.13 indicates the results of the non-invariance as per each path.

Results show that one path is significantly different between the two educational level groups, namely the Process - Effort path, but that the other causal relationships were not supported (as shown in Table 6.13). The regression estimates for individuals with higher levels of education are stronger than those for individuals with lower levels of education.

Therefore, the hypothesised education moderating effect H17 is partially supported for the South African sample.

As shown in Table 6.9 the following hypothesis was supported on both female and male:

H17a: Educational level positively moderates the relationships between effort expectancy factors and success of agile software development projects.

As shown in Table 6.9 the following hypothesis were supported on both female and male:

H17b: Educational level positively moderates the relationships between process factors and performance expectancy factors for the success of agile software development projects.

H17c: Educational level positively moderates the relationships between process factors and effort expectancy factors for the success of agile software development projects.

Table 6.9: South Africa's un-standardised estimates and critical ratio for education groups

Path			Diploma and less		Degree and above		Notes
			Estimate	C.R.	Estimate	C.R.	
Performance	<---	Project	-0.042	-2.192*	-0.02	-0.987 n/s	Path not sig. for degree and above
Effort	<---	Process	0.038	0.692 n/s	0.235	3.62***	Path not sig. for diploma and less
Effort	<---	Tech	0.188	2.046*	0.068	0.688 n/s	Path not sig. for degree and above
Effort	<---	Org	0.555	6.373***	0.487	6.778***	
Performance	<---	Process	-0.016	-0.722 n/s	0.071	2.341*	Path not sig. for diploma and less
Performance	<---	People	0.047	2.531*	0.049	2.22*	
Performance	<---	Org	0.739	20.339***	0.702	20.761***	
Intention	<---	Effort	0.166	3.066**	0.088	1.567 n/s	Path not sig. for degree and above
Intention	<---	Performance	0.658	6.542***	0.757	8.53***	
Actual	<---	Effort	0.404	7.75***	0.264	6.137***	
Actual	<---	Performance	0.053	0.656 n/s	0.24	4.057***	Path not sig. for diploma and less
Actual	<---	Process	0.214	5.347***	0.259	6.607***	

Note: *** p<. 001; ** p<. 01; * p<. 05

Table 6.10: Multiple group covariance structure analysis for education groups

Model	**DF**	**CMIN**	**P**	**NFI Delta-1**	**IFI Delta-2**	**RFI rho-1**	**TLI rho2**
Measurement weights	12	16.076	.188	.011	.001	.001	.001
Structural weights	7	23.032	.000	.367	.002	.003	-.014

Table 6.11: Baseline comparisons for South African education groups

Model	**NFI Delta1**	**RFI rho1**	**IFI Delta2**	**TLI rho2**	**CFI**	**ΔCFI**
Unconstrained	.989	.963	1.004	1.014	1.000	
Structural weights	.978	.952	1.001	1.002	1.000	0.000
Structural co-variances	.947	.924	.981	.972	.981	0.019
Structural residuals	.931	.911	.969	.959	.968	0.013
Saturated model	1.000		1.000		1.000	
Independence model	.000	.000	.000	.000	.000	

Table 6.12: Scalar means estimates for South African education groups

	Estimate	**S.E.**	**C.R.**	**P**
Org	3.523	.030	-116.387	***
Process	2.136	.049	-43.604	***
People	3.172	.061	-51.975	***
Tech	.826	.027	-30.522	***
Project	3.081	.060	-51.770	***

Table 6.13: Path equivalence verification with difference in chi-square for South Africa education groups

Path			Diploma and less		Degree and above		Invariance		
			Estimate	**C.R.**	**Estimate**	**C.R.**	**Difference χ^2**	**Difference df**	**p**
Performance	<---	Project	-0.042	-2.192*	-0.02	-0.987 n/s	0.006	1	0.848
Effort	<---	Process	0.038	0.692 n/s	0.235	3.62***	0.196	1	0.051
Effort	<---	Tech	0.188	2.046*	0.068	0.688 n/s	-0.057	1	0.794
Effort	<---	Org	0.555	6.373***	0.487	6.778***	0.657	1	0.657
Performance	<---	Process	-0.016	-0.722 n/s	0.071	2.341*	0.100	1	0.200
Performance	<---	People	0.047	2.531*	0.049	2.22*	0.009	1	0.748
Performance	<---	Org	0.739	20.339***	0.702	20.761***	-0.037	1	0.556
Intention	<---	Effort	0.166	3.066**	0.088	1.567 n/s	-0.057	1	0.509
Intention	<---	Performance	0.658	6.542***	0.757	8.53***	0.092	1	0.546
Actual	<---	Effort	0.404	7.75***	0.264	6.137***	0.019	1	0.760
Actual	<---	Performance	0.053	0.656 n/s	0.24	4.057***	0.174	1	0.106
Actual	<---	Process	.214	5.347***	0.259	6.607***	0.039	1	0.579

Note: *** p<. 001; ** p<. 01; * p<. 05

6.3.4 Experience influence

The experience variable shows that 75.0% of the respondents had zero to four (0-4) years' experience with agile software projects and 25% had more than 2 years' experience. Thus the experience variable was divided into two groups, that of 4 years and less (regarded as group A), and that of above 4 years (regarded as group B), to fit the explanation of the moderating effect.

The group A sample fit statistics, results are:

- CMIN = 2.936 with d.f =6 and CMIN/d.f ratio = 0.326;
- SRMR = 0.010;
- CFI =0.998; and
- RMSEA =0. 000 with 90 percent confidence interval (0.000 and 0.000) and PCLOSE =0. 998.

The group B sample fit statistics are:

- CMIN = 6.708 with d.f =5 and CMIN/d.f ratio = 1.342;
- SRMR = 0.042;
- CFI = 0.981; and
- RMSEA = 0.055 with 90 percent confidence interval from 0.000 to 0.149 and PCLOSE = 0.393

The study thus established an acceptable model fit for experience samples. A group multiple analysis test for factor structure equivalence was computed for the two groups concurrently rather than distinctly. The estimation results and critical ratio are shown in Table 6.14.

All paths are significant for both experience groups, except Performance-Project, Effort-Process, Performance-People and Actual-Performance for the group with more than 4 years' experience.

The AMOS v23 model comparison results revealed that the two experience groups are invariant. Supposing the unconstrained model to be correct, the measurement weights between the two experience groups is invariant as is shown by the non-significant p value which is greater than 0.05 (Table 6.15).

Also the two experience groups are invariant based on the structural weights, assuming the measurement model to be correct (as shown in Table 6.15). The interpretation indicates a non-significant p value which is greater than 0.05, which shows invariance between the two groups.

Table 6.14: South Africa's un-standardised estimates and critical ratio for experience groups

Path			4 Years and less		More than 4 Years		
			Estimate	C.R.	Estimate	C.R.	Notes
Performance	<---	Project	-0.05	-3.094**	0.021	0.755n/s	Path not sig. for More than 4 Years
Effort	<---	Process	0.149	3.326***	-0.022	-0.281 n/s	Path not sig. for More than 4 Years
Effort	<---	Org	0.516	8.433***	0.559	4.604***	
Performance	<---	People	0.047	2.785**	0.031	1.148 n/s	Path not sig. for More than 4 Years
Performance	<---	Org	0.74	27.178***	0.704	14.405***	
Intention	<---	Effort	0.091	2.017*	0.205	2.616**	
Intention	<---	Performance	0.726	9.859***	0.711	4.912***	
Actual	<---	Effort	0.32	8.038***	0.338	5.515***	
Actual	<---	Performance	0.164	2.972*	0.18	1.845 n/s	Path not sig. for More than 4 Years
Actual	<---	Process	0.258	7.698***	0.175	3.495***	

Table 6.15: Multiple group covariance structure analysis for experience groups

Model	DF	CMIN	P	NFI Delta-1	IFI Delta-2	RFI rho-1	TLI rho2
Measurement weights	10	13.050	.221	.009	.010	.012	.012
Structural weights	8	10.863	.249	.000	.009	.002	.003

As shown in Table 6.14 the following hypothesis was supported on both female and male:

H16a: Experience positively moderated the relationships between effort expectancy factors and success factors of the agile software development projects.

As shown in Table 6.14 the following hypothesis were not supported on both female and male:

H16b: Experience positively moderated the relationships between process factors and performance expectancy factors of the agile software development projects.

H16c: Experience positively moderated the relationships between process factors and effort expectancy factors of the agile software development projects.

In the study it was assumed that the factor loadings are equal between the two groups. Therefore, assuming as well that the structure weights are correct, the structural residuals are invariant between the two groups, depicted by the non-significant p value results. Due to the fact that several tests have been done to end up with the structural weights, and assuming that examining residual invariant is more stringent than necessary, the residual invariance is not applied in this study.

The model fit results for the multiple group analysis discovered acceptable outputs with respect to changes in CFI (less than 0.01), as depicted in Table 6.16. Consequently, it is concluded that the model is functioning homogeneously across group A and group B.

Table 6.16: Baseline comparisons for South African experience groups

Model	NFI Delta1	RFI rho1	IFI Delta2	TLI rho2	CFI
Unconstrained	.993	.979	1.006	1.021	1.000
Structural weights	.984	.968	1.004	1.009	1.000
Structural co-variances	.982	.973	1.010	1.014	1.000
Structural residuals	.979	.974	1.012	1.016	1.000
Saturated model	1.000		1.000		1.000
Independence model	.000	.000	.000	.000	.000

An important consequence of establishing measurement weights invariance is the possibility of carrying out the subsequent tests of invariance, differences in structural weights and latent means. The level of examination included an approximation of means on latent and observed variables through use of covariance and mean structure analyses (Pomykalski et al., 2008; Rohani et al., 2009). Using the recommendations stated previously in 6.2, and using AMOS graphics v23.0, while constraining group A to be the reference group, the scale invariance indicates that group B mean scores are positive, meaning that group B has higher mean scores than the group A reference group.

Table 6.17: Scalar means estimates for South African experience groups

	Estimate	**S.E.**	**C.R.**	**P**
Org	2.523	.070	145.387	***
Process	1.136	.069	54.703	***
People	4.172	.048	55.864	***
Tech	3.886	.056	40.533	***
Project	4.084	.060	52.110	***

The results indicate that all the mean scores of the main variables are higher for group A than they are for group B and the differences are significant, since the p values are less than 0.05. Based on the p values and critical ratios, as indicated in Table 6.17, the non-reference group mean scores are statistically different from the reference group in respect to Org (Organisational factors), Process (Process factors), People (People factors), Tech (Technological factors) and Project (Project factors).

Chen et al. (2005) recommended the use of multiple fit indices for evaluating the model goodness of fit in explaining the outputs of mean and covariance structure, which comprise SRMR, CFI, and RMSEA. The model fit statistics outputs for these indices in relation to the experience moderator all indicate a good fit:

- CFI=1.000;
- SRMR= 0.010 and
- RMSEA =0. 000 with 90 percent confidence interval (0.000 and 0.038) and PCLOSE = 1.000.

Lastly, the invariant output at the structural weight level indicates that factor loadings for the structural paths are not significantly different between group A and B. Hence, the hypothesised moderating influence of experience H16 is not supported for the South African sample. It is clear that, when team members in project are inexperienced, skill transfer becomes not significant to ensure the quality of project results.

6.4 Summary

This chapter explained the analysis and evaluation of the research framework in relation to the moderating variables. The demographics of the respondents and effects of moderators were discussed. Relying on the outcomes of the structural equation model in Figure 6.2, a framework was established that was confirmed and presumed as the closest fit for South Africa data samples. Furthermore, multi-group analysis was used to find the effects of moderating factors on the conceptual framework constructs or decision variable results of which were used to prove the hypotheses.

CHAPTER 7: DISCUSSION OF RESULTS

7.1 Introductions

This chapter explains the outline, discussion and interpretation of findings. Findings of the hypotheses are discussed in relation to their relevance to framework and practice. The research questions are re-examined to assess the degree to which the research objectives have been accomplished.

The primary research question of this study was to identify and provide insight into the critical success factors that influence the success of agile software development projects. The inspiration of this study was the fact that, despite of the several research studies on agile software development projects, there is still a serious shortage of scientific assessment regarding the critical success factors for agile software development projects. The study was further inspired by difficulty identifying relevant literature and theoretical frameworks when several search engine such EBSCO host and Google scholar were researched, and also because there is increasing failure of agile software development projects. Thus the researcher sought to answer the research questions, with South Africa as a sample population.

Agile methodologies were designed to resolve concerns about traditional software development practices which have proven to be problematic (Misra, Kumar & Kumar, 2006). Although many agile professionals and agile management have received training with regard to the implementation of agile software development projects, they still struggle to appreciate the value thereof and some of them preferred the traditional software development projects (Chow & Cao, 2008). On the other hand many agile professionals lack training (Misra et al., 2009). In South Africa, it has previously been found that some provinces outsourced training to private consultants who themselves had insufficient knowledge and practical training experience (Misra et al., 2009).

Recently, researchers have cited the key problems in implementing agile software development projects as being that the relevant software development professionals have not been effectively trained to handle the changes, while the software projects experts do not adequately discourse the main difference between traditional and agile software development projects (Chow & Cao, 2008).

Many organisations are still confused about which methodology should be used to develop software projects. There exist several software development methodologies and critical success factors, and software development professionals and their management find it difficult to select the most appropriate methodology (Nguyen, 2016). This is exacerbated by the fact that agile professionals tend to be passionately devoted to agile and will advocate for its use without considering organisational and other factors (Bossini & Fernández, 2013).

Chow and Cao (2008) have confirmed the importance of using a modelling technique to identify the critical success factors of agile software development projects. There remains a gap in literature that this research aims to fill through exploring organisations' views and opinions concerning agile software development projects in South Africa and how they perceive the critical success factors of their organisation. This calls for the development of a theoretical framework that exhaustively determines the critical success factors that influence the implementation of agile software development projects using agile methodologies.

This led the researcher to ask the following primary research question: What are the critical success factors that influence the success of software development projects using agile methodologies? Due to the integrated and complicated nature of agile, three sub-questions needed to be examined in order to appropriately answer the primary question. These were:

1. How do agile professionals perceive the adoption of the agile software development projects in South Africa

2. What is the most appropriate theoretical framework which can be adapted to model the critical success factors of agile software development projects?

3. How can critical success factors be structured into a framework that can inform agile professionals and the community?

The primary objective of this study is to identify and provide insight into the critical success factors that influence the success of software development projects using agile methodologies. To accomplish this, the following specific objectives were pursued:

1. To determine how agile software development project success is perceived and evaluated within organisations in South Africa;

2. To determine the most appropriate theoretical framework which can be adapted to model the critical success factors;

3. To construct a structural equation model for the critical success factors of agile software development projects.

Based on the literature review (chapter 2) and the theoretical framework and research model adopted (as discussed in chapter 3), the research study followed a succession of stages by applying mixed research methodologies as discussed in detail in chapter 4. The analysis approach and interpretation of results was discussed in chapter 5, while the effect of moderating factors was discussed in chapter 6. Below, the various hypotheses are discussed in light of the results, as well as the implications of these findings.

7.2 Research discussion of the results and findings

7.2.1 Findings in the hypotheses

Chapter 4 of the study explained the hypotheses that needed to be examined to answer the research questions for this study. Table 7.1 shows the summary of which hypotheses were supported or not supported in the research study.

Table 7.1: Findings of the research hypotheses

Proposed hypotheses	Findings
H1: Social influences have a positive effect on intention of the agile methodologies for the success of an agile software development projects.	Not supported
H2: Performance expectancy has a positive effect on agile intention for the success of an agile software development projects.	Supported
H3: Effort expectancy has a positive effect on agile intention for the success of the agile software development projects.	Supported
H4: Behavioural intention has a positive effect on the success factors of an agile software development projects.	Not Supported
H5a: Technological factors have a positive effect on behavioural intention for the success of agile software development projects.	Not Supported
H5b: Technological factors have a positive effect on effort expectancy for the success of the agile software development projects.	Supported
H5c: Technological factors have a positive effect on performance expectations for the success of agile software development projects.	Not Supported
H5d: Technological factors have a positive effect on the success of the agile software development projects	Not Supported

Proposed hypotheses	Findings
H11a: Process factors have a positive effect on effort expectancy for the success of the agile software development projects.	Supported
H11b: Process factors have a positive effect on performance expectations for the success of the agile software development projects.	Supported
H11c: Process factors have a positive effect on behavioural intention for the success of the agile software development projects.	Not Supported
H11d: Process factors have a positive effect on the success of the agile software development projects.	Supported
H12a: Project factors have a positive effect on performance expectations for the success of the agile software development projects.	Partially Supported
H12b: Project factors have a positive effect on effort expectancy for the success of the agile software development projects.	Not Supported
H12c: Project factors have a positive effect on behavioural intention for the success of the agile software development projects.	Not Supported
H12d: Project factors have a positive effect on the success of the agile software development projects.	Not Supported
H13a: Performance expectancy has a positive effect on the success of the agile software development projects.	Supported
H13b: Effort expectancy has a positive effect on the success of the agile software development projects.	Supported
H13c: Social Influence has a positive effect on the success of the agile software development projects.	Not Supported
H6a: Culture have a positive effect on the success of the agile software development projects.	Not Supported
H6b: Political have a positive effect on the behavioural intention success of the agile software development projects.	Not Supported
H6c: Political factors have a positive effect on the organisational success factors of the agile software development projects.	Not Supported
H7a: Political factors have a positive effect on agile professionals' social influence for the success of the agile software development projects.	Not Supported
H7b: Political factors have a positive effect on agile professionals' effort expectancy for the success of the agile software development projects.	Not Supported
H8a: People factors have a positive effect on effort expectancy factors for the success of the agile software development projects.	Not Supported

Proposed hypotheses	Findings
H8b: People factors have a positive influence on performance expectancy factors for the of the agile software development projects.	Supported
H8c: People factors have a positive influence on social influence factors for the success of the agile software development projects.	Not Supported
H8d: People factors have a positive effect on the overall perceived success of the agile software development projects.	Not Supported
H9a: Culture has a positive effect on social influences for the success of the agile software development projects.	Not Supported
H9b: Culture has a positive effect on behavioural intention for the success of the agile software development projects.	Not Supported
H10a: Organisational factors have a positive effect on effort expectancy for the success of the agile software development projects.	Supported
H10b: Organisational factors have a positive effect on performance expectancy for the success of the agile software development projects.	Supported
H10c: Organisational factors have a positive effect on the success of the agile software development projects.	Not Supported
H14: Gender moderates the relationships among the proposed model constructs	Not Supported
H15: Age moderate the relationships among the proposed model constructs	Not Supported
H16: Experience moderates the relationships among the proposed model constructs	Not Supported
H17: Educational level moderates the relationships among the proposed model constructs	Supported
H14a: Gender positively moderates the relationships between individual performance expectancy factors and overall success of agile software development projects.	Not supported
H14b: Gender positively moderates the relationships between individual effort expectancy factors and overall success of agile software development projects.	Supported
H14c: Gender positively moderates the relationships between project factors and performance expectancy factors for the success of agile software development projects.	Not supported
H14d: Gender positively moderates the relationships between people factors and performance expectancy factors for the success of agile software development projects.	Not supported

Proposed hypotheses	Findings
H15a: Age positively moderates the relationships between process factors and performance expectancy factors for the success of agile software development projects.	Not supported
H15b: Age positively moderates the relationships between performance expectancy factors and intention factors for the success of agile software development projects.	Supported
H15c: Age positively moderates the relationships between effort expectancy factors and intention factors for the success of agile software development projects.	Supported
H16a: Experience positively moderated the relationships between effort expectancy factors and success factors of the agile software development projects.	Supported
H16b: Experience positively moderated the relationships between process factors and performance expectancy factors of the agile software development projects.	Not supported
H16c: Experience positively moderated the relationships between process factors and effort expectancy factors of the agile software development projects.	Not supported
H17a: Educational level positively moderates the relationships between effort expectancy factors and success of agile software development projects.	Supported
H17b: Educational level positively moderates the relationships between process factors and performance expectancy factors for the success of agile software development projects.	Not supported
H17c: Educational level positively moderates the relationships between process factors and effort expectancy factors for the success of agile software development projects.	Not supported

7.2.2 Discussion and interpretation of the hypotheses

7.2.2.1 Factors that fail to influence Behavioural Intention

Social influence has a non-significant influence on behavioural intention factors relating to the success of an agile software development projects. The expectation that social influence would have a positive effect (H1) was not supported. The results showed that organisation experts, such as software developers and team leaders, do not successfully influence, motivate or encourage their team members to use agile methods. Neither do management nor friends successfully encourage team members to use agile methods. It was thus not expected that team members would use agile methodologies. In addition, most people whom the team members value in their lives do not use agile methodologies and most people do not base their good opinion of team members on whether they use agile methodologies.

These results are in agreement with Misra et al's (2012) conclusion that continued usage of traditional software development success factors in agile software projects may turn obligatory usage into voluntary usage therefore making social effects non-significant.

Furthermore, the following hypothesis which expected a positive effect on behaviour intention were also not supported, namely: H5a, H11c, H12c, H13c, H6b, H6c, H7a, H8c, H9a, and H9b.

7.2.2.2 Factors influencing Behavioural Intention

The findings show that performance expectancy factors do have a positive effect on behavioural intention factors for the success of an agile software development projects (H2), thus this hypothesis was supported. The findings are in line with Curtis and Payne (2008) who found that agile approaches enable team members to accomplish tasks more efficiently, usefully, and productively, and also increase their chances of getting promotion, thus encouraging them to plan, make an effort and intend to practice agile software projects on regular basis.

Effort expectancy factors also have a positive effect on behavioural intention (H3), thus this hypothesis was also supported. The findings are in line with Curtis and Payne (2008) who found that agile approaches enable team members to accomplish interaction tasks clearly, understandably, and easily, thus encouraging them to plan, make an effort and intend to practice agile software projects on regular basis.

Because the success factors of agile software projects are integrative in nature, team members have a high anticipation for the success of agile projects. Therefore, management must established realistic expectations for inexperienced team members so that they work and benefit at the same time as the experienced team members.

7.2.2.3 Factors influencing Effort Expectancy

Agile methods are known to be complex, and team members who perceive agile to be easy, clear and understandable will have increased benefits, therefore gaining self-efficacy which is key. Team members acquire experience of agile methods with time. The implications of this is that team members with positive effort expectancy are able to accomplish important aspects of the agile methodology including following agile project management style, maintaining a strong communication etc.

Process factors have a significantly positive effect on effort expectancy factors for the success of agile software development projects (H11a). The hypothesis was supported by the findings of the research study. The results of the research show that agile user's effort expectancy is dependent on:

- following, agile project management style;
- following agile-oriented requirements processes;
- clearly defining project scope and objectives;
- honoring a regular working schedule;
- following an agile-friendly progress tracking mechanism;
- using flexible time-boxing or rapid-pace progress measurement techniques instead of document milestones or work breakdown structure;
- maintaining a strong communication focus and rigorous communication schedule;
- maintaining an agile-oriented configuration management process;
- ensuring strong customer commitment and presence, including a customer representative on the project with full authority and knowledge to make decisions on-site; and
- working in a facility with proper agile-style work environment.

Similarly, *organisational factors* have a significantly positive effect on effort expectancy factors (H10a). The results show that an Agile user's effort expectancy to practice agile methods is highly dependent on the project team working in a facility with proper agile-style work environment, including:

- all team members working in the same location for ease of communication
- casual, constant contact;
- a cooperative organisational culture instead of an hierarchal one;
- an oral culture placing high value on fluid, face-to-face communication;
- a reward system appropriate for agile;
- universal accepted of agile in the organisation;
- a committed sponsor or a committed organisation manager; and
- strong management or executive support.

Technological factors also have a significantly positive effect on effort expectancy factors for agile software development project success (H5b). The hypothesis was supported by the findings of the research study. The outcomes indicate that agile user's effort expectancy factors to practice agile methods is highly dependent on:

- a well-defined coding standard imposed up front;
- pursuit of simple design;
- pursuit of vigorous refactoring activities;
- maintenance of the right amount of documentation;
- correct integration testing;
- delivery of most important features first, and
- appropriate technical training to team members.

Project factors also have a significantly positive effect on effort expectancy factors for agile software development project success (H12b). The results revealed that agile user's effort expectancy to practice agile methods is highly dependent on having a project that:

- has a dynamic, accelerated project schedule;
- has variable scope with emerging requirements;
- involves non-life-critical software (although it could be business mission-critical software);
- has no multiple, independent teams working together;
- has up-front, detailed cost evaluation completed and approved; and
- has small teams.

People factors also have a significantly positive effect on effort expectancy factors (H8a). The results revealed that agile user's effort expectancy to practice agile methods is highly dependent on:

- a well-defined coding standard that is imposed up front;
- simple design;
- vigorous refactoring activities to ensure the results are optimal and to accommodate all changes in requirements;
- the right amount of documentation for agile purpose;
- continuous and rigorous unit and integration testing strategies for each and every iteration;
- delivery of working software regularly within short periods of time;

- delivering the most important features first;
- employing proper platforms, technologies, and tools suitable to agility practice; and
- appropriate technical training to the team, including training on subject matter and agile processes.

7.2.2.4 Factors influencing Performance Expectancy

Project factors have a positive effect on performance expectancy factors for agile software development project success which was significant (H12a). The results show that agile user's performance expectancy to practice agile methods is highly dependent on:

- the dynamic, accelerated schedule;
- a variable scope with emerging requirements;
- projects that involve a non-life-critical software; (although it could be business mission-critical software);
- no multiple, independent teams working together;
- up-front, detailed cost evaluation completed and approved; and
- small team size.

Process factors also have a significant positive effect on performance expectancy factors (H11b). The outcomes show that agile user's performance expectancy to practice agile methods is highly dependent on:

- agile project management style;
- agile-oriented requirements processes;
- project scope and objectives which are well-defined;
- a regular working schedule;
- an agile-friendly progress tracking mechanism;
- flexible time-boxing or rapid-pace progress measurement techniques instead of document milestones or a work breakdown structure;
- a strong communication focus and rigorous communication schedule;
- an agile-oriented configuration management process; and
- a strong customer commitment and presence with a customer representative on the project with full authority and knowledge to make decisions on-site.

People factors have a significant positive effect on performance expectancy factors for agile software development project success (H8b). The results showed that agile user's performance expectancy to practice agile methods is highly dependent on:

- a well-defined coding standard being imposed up front;
- simple design;
- vigorous refactoring activities to ensure the results are optimal and to accommodate all changes in requirements;
- maintaining the right amount of documentation for agile purpose;
- continuous and rigorous unit and integration testing for each and every iteration;
- delivering working software regularly within short periods of time;
- delivering most important features first;
- using proper platforms, technologies, and tools suitable for agility practice; and
- providing appropriate technical or technological training to the team, including training on subject matter and agile processes.

Technological factors have a positive effect on performance expectancy factors for agile software development project success (H5c). The outcomes indicate that agile user's performance expectancy factors to practice agile methods is highly dependent on:

- a well-defined coding standard being imposed up front;
- pursuit of simple design;
- vigorous refactoring activities;
- maintenance of the right amount of documentation;
- correct integration testing;
- delivery of the most important features first; and
- appropriate technical training to team members.

Organisational factors have a positive effect on performance expectancy factors for agile software development project success which was significant (H10b). The results show that agile user's performance expectancy to practice agile methods is highly dependent on:

- the project team working in a facility with proper agile-style work environment;
- all team members working in the same location for ease of communication;
- casual, constant contact;
- a cooperative organisational culture instead of an hierarchal one;

- an oral culture placing high value on fluid, face-to-face communication;
- a reward system that is appropriate for agile;
- agile methodologies being universally accepted in the organisation;
- a committed sponsor or a committed organisation manager; and
- strong management or executive support.

Factors influencing Actual Success of agile software projects

The research shows that *Behavioural intention* factors have a non-significant positive effect on the actual success of agile software development projects, thus hypothesis H4 is not supported. This is in line with previous research by Venkatesh et al, (2003) which does not supported the validity of this hypothesis and which was confirmed by Tibenderana and Ogao (2008). Those who does not intend to practice the agile method in order to deliver a good product or good software results are more likely not to meet all requirements and objectives, does not deliver software projects on time and does not deliver software projects within estimated effort and cost of software projects.

Performance expectancy factors were also found to have a positive effect on actual success of agile software development projects, thus hypothesis (H13a) is supported. The findings are in line with Venkatesh et al's (2003) findings which show that agile practitioners who expect to accomplish tasks more efficiently, usefully, and productively and believe they have increased chances of getting promotion through using agile methods, are more likely to deliver a good product or good software results, meet all requirements and objectives, deliver software projects on time and deliver software projects within estimated effort and cost.

Process factors were also found to have a significant positive effect on the actual success of agile software development projects, thus hypothesis (H11d) was also supported. Again this is in line with Venkatesh et al's (2003) findings, which were examined and confirmed true by Tibenderana and Ogao (2008). When agile processes are followed, the team is more likely to deliver a good product or software results, meet all requirements and objectives, deliver software projects on time and deliver software projects within estimated effort and cost.

Finally, the research found that *effort expectancy* factors also have a positive effect on actual success for the success of an agile software development projects, supporting hypothesis (H13b). These findings are also consistent with Venkatesh et al. (2003). When team members expect that agile methodologies are clear and understandable and will enable them to accomplish tasks easily, they are more likely to deliver a good product or software results, meet all requirements and objectives, deliver software projects on time and deliver software projects within estimated effort and cost.

7.2.3 Discussion of the hypotheses in relation to moderating factors

Hypotheses H14 to H17 suggest the effects of moderators on the hypothesised relationships between the constructs or decision variables. The hypotheses relate to four moderating factors namely: age, gender, experience and level of education.

Gender influence

The measurement invariance outputs show the difference in CFI between the male and female groups, which does not exceed the 0.01. The latent means score differences between the two gender groups was tested after establishing measurement invariance. Constraining the female group to be the reference group, the scale invariance indicated that the male group's mean scores are positive, meaning that the male group has higher mean scores compared to the female reference group.

The results indicate that all the mean scores of the main variables are higher for males than they are for females and the differences are significant since the p value is less than 0.05. Based on the p values and the critical ratio, as shown in Table 6.4, the non-reference group mean scores are statistically different from the reference group in respect to Org (Organisational), Process (Process factors), People (People factors), Tech (Technological factors) and Project factors. The invariant output at the structural weight level indicates that factor loadings for the structural paths are not significantly different between females and males.

Hence, the hypothesised moderating influence of gender H14 is not supported for the South African sample. This implies that gender did not have an influence on the research study.

Age influence

The measurement invariance outputs show the difference in CFI between age groups, and these do not exceed 0.01. The latent means score differences between the age groups was tested after establishing the measurement invariance.

The results show the narrow range for age groups which makes the comparison ineffective. The age groups are almost the same, which leads to non-invariance results in the structural equation models. Computing the mean and covariance structure analysis when constraining the age group

30 and above to be the reference group, the scalar mean scores indicates that the younger group (age 30 and below) has lower mean scores than the older group. Moreover, the two age groups statistically differ on mean scores for the latent variables Effort and Performance, although, they do not differ significantly on Intention and Actual. The result of invariance at the structural weight level indicates that the structural paths are not significantly different between the two age groups.

Thus, the hypothesised age moderating effect H15 is not supported for the South African sample, indicating that age did not influence the research study.

Education influence

The measurement invariance outputs show the difference or changes in CFI between education groups and these do not exceed 0.01. The structural weights outputs shows changes in the CFI, which do exceed the 0.01, thus not supporting the structural invariance outputs.

The latent means score differences between the two educational groups was checked after establishing measurement invariance. Computing the mean and covariance structure analysis while constraining the group “education on diploma and above” to be the reference group, the scalar mean scores indicates that the education level below diploma has a negative effect. This shows that the reference group (diploma and above) has higher mean scores than the diploma and below group.

The invariant paths were evaluated one path at a time in order to discover the non-equality between the groups. Model invariance between the two groups is acknowledged through use of model fit measurements. The invariant outcome at the structural weights level shows that the structural paths are significantly different between the two levels of education. The total effect results show that individuals with less education depend on their experience in making decisions.

The hypothesised education moderating effect H17 is partially supported for the South African sample and this has a significant effect to the research model:

- Level of education positively moderated the relationships between effort expectancy factors and actual success factors in agile software development projects (H17a)
- Level of education positively moderated the relationships between process and performance expectancy factors in agile software development projects (H17b); and
- Level of education positively moderated the relationships between process and effort expectancy factors in agile software development projects (H17c).

Level of experience influence

The two experience groups are invariant based on the structural weights, assuming the measurement model to be correct. The interpretation shows a non-significant p value which is greater than 0.05, which shows invariance between the two groups. In the study it was assumed that the factor loadings are equal between the two groups. Therefore, assuming as well that the structure weights are correct, the structural residuals are invariant between the two groups, depicted by the non-significant p value results.

The results indicate that all the mean scores of the main variables are higher for group A (4 years and less) than they are for group B (more than 4 years) and the differences are significant since the p value is less than 0.05. Based on the p values and critical ratio, as indicated in Table 6.16, the non-reference group mean scores are statistically different from the reference group A in respect to Org, Process, People, Tech and Project. The invariant output at the structural weight level indicates that factor loadings for the structural paths are not significantly different between group A and B.

The hypothesised moderating influence of level of experience (H16) is thus not supported for the South African sample.

7.2.4 Conclusion Related to the Hypotheses

Results of this study confirm the previous findings by several researchers that organisational, process, people and project factors are critical success factors and have an influence on performance expectancy factors in the final model. The findings thus support recommendations made by Misra et al. (2012). Further, the research highlighted that level of education factors influence other variables in the research framework, meaning that the more educated you are, the more likely you are to have a positive orientation towards the critical success factors influencing agile project success.

Based on these findings, company management and agile practitioners should train and be aware of organisational factors as the main contributing factors in the study which could affect success of agile projects, particularly in the agile software engineering research area.

7.3 Findings and discussions in connection to the research questions

The findings and discussions in relationship to the research questions are explained below.

7.3.1 The research question

The primary research question was:

What are the critical success factors that influence the success of agile software development projects using agile methodologies?

In answer to this, the research has revealed that the most influential factors are as follows:

- Organisational factors are relevant critical success factors that influence performance expectancy factors (H10b; β=0.721; $p<0.05$) and these were the most influential factors in the research model.
- Performance expectancy factors are relevant critical success factors that influence intention factors (H2; β=0.705; $p<0.05$) and these were the second most influential factors in the research study model.
- Organisational factors are relevant critical success factors that influence effort expectancy factors (H10a; β=0.528; $p<0.05$) and these were the third most influential factors in the research model.
- Effort expectancy factors are relevant critical success factors that influence actual success factors (H16; β=0.316; $p<0.05$) and these were the fourth most influential factors in the research study model.

In terms of influence on performance expectancy factors, organisation factors were the most influential (H10b; β=0.721; $p<0.05$), followed by project factors (H12a; β=-0.041; $p<0.05$), process factors (H11b; β=0.036; $p<0.05$), and people factors (H8b; β=0.047; $p<0.05$). Technological factors did not affect performance expectancy factors (H5c; β=0.047; $p<0.05$) (see Table 5.19 and Figure 5.7).

In terms of influence on effort expectancy factors, organisational factors were the most relevant critical success factors (H10a; β=0.528; $p<0.05$), followed by process factors (H11a; β=0.132; $p<0.05$), and technological factors (H5b; β=0.185; $p<0.05$). Project factors (H12b; β=-0.040; $p<0.05$) and people factors (H8a; β=0.012; $p<0.05$) did not affect effort expectancy factors (see Table 5.19 and Figure 5.7).

In terms of intention factors, the most influential factors were performance expectancy factors (H2; β=0.705; $p<0.05$). Effort expectancy factors (H3; β=0.138; $p<0.05$) also affect intention factors (see Table 5.19 and Figure 5.7).

With regards to actual success, effort expectancy factors were the most relevant critical success factors (H16; β=0.316; $p<0.05$) followed by process factors (H11d; β=0.230; $p<0.05$) and performance expectancy factors (H13a; β=0.132; $p<0.05$). Intention factors did not affect actual success factors (H4; β=0.065; $p<0.05$) (see Table 5.19 and Figure 5.7).

Figure 5.7 seeks to explain the structural equation model of the study that could be used to explain the use of and success of agile software development projects. The results of the structural equation modelling are shown in Tables 5.18 and 5.19. The results of the hypotheses show how each path in the model is established.

7.3.2 Construct formulation

In the literature review, 47 factors were recognised and categorised into nine groups: organisational factors, process factors, people factors, technological factors, project factors, performance expectancy factors, effort expectancy factors, intention factors and actual success factors. The factors were obtained from the literature review and interviews with agile management and practitioners. The factors were assessed using content analysis and 47 factors were obtained. By responding to the questionnaire (see Appendix B), agile management and practitioners were requested to critic the relevance and importance of these factors.

7.3.3 Conclusions in connection to the research questions

Results from the research questions illustrate that the most influential critical success factors with regards to agile software development projects relate to the influence organisational factors have on performance expectancy factors. This suggests that companies or organisations must be ready to take the principal responsibility for ensuring effective practise of agile software development projects. Management staff need to support team members and agile experts by providing training and promoting good and rapid communication, trusting workmates, receiving necessary feedback and input from customers, practicing agile methodologies, having a customer-centric organisation, encouraging changing requirements and having a co-operative management structure.

The important role moderating factors can play must be recognised and accepted. This research study discovered that education level is the key moderating factor in the South African context, with a significant moderating effect on the research framework. Meanwhile age, level of experience and gender were not significant in the model. Therefore, companies need to focus on educating team members, should offer training courses and should employ qualified staff who are well educated to suit the job.

7.4 The modified research model (Final research model)

The primary objective of the study was to identify and provide insight into the critical success factors that influence the success of software development projects using agile methodologies. This was accomplished through the secondary objectives:

1. *To determine how agile software development project success is perceived and evaluated within organisations in South Africa;*

2. *To determine the most appropriate theoretical framework which can be adapted to model the critical success factors; and*

3. *To construct a structural equation model for agile software development project success factors.*

Objective 2 was achieved and the theoretical framework is presented in chapter 4 and the model shown in Figure 7.1 below.

The third objective was achieved and the structural model has been presented in detail in chapter 6 and 7, and the final model is shown below (figure 7.1). This version takes into account the results of the hypothesis testing, specifically the most significant findings that:

- organisational factors were found to have significant influence on performance expectancy factors (H12b);
- organisation factors were found to have significant influence on effort expectancy factors (H10a),
- performance expectancy factors were found to have significant influence on intentional factors (H2) ; and
- effort expectancy factors were significant to actual success factors.

Of the remaining hypotheses, some were supported while others were not, as shown in Table 7.1.

As for the moderating effect factors, the following causal relationship were eliminated:

- all age moderating effects hypotheses (H15a, H15b and H15c),
- all gender moderating effects hypotheses (H14a, H14b H14c and H14d), and
- all level experience moderating effects hypotheses (H17a, H17b and H17c).

Education moderating effects hypotheses (such as H16) were found to be significant and these were discussed in detail in chapter 6.

The resulting model, which was originally constructed based on the literature and refined by this research, can act as a guideline for agile management, team members and agile experts, as well as other companies or organisations that use agile methodologies.

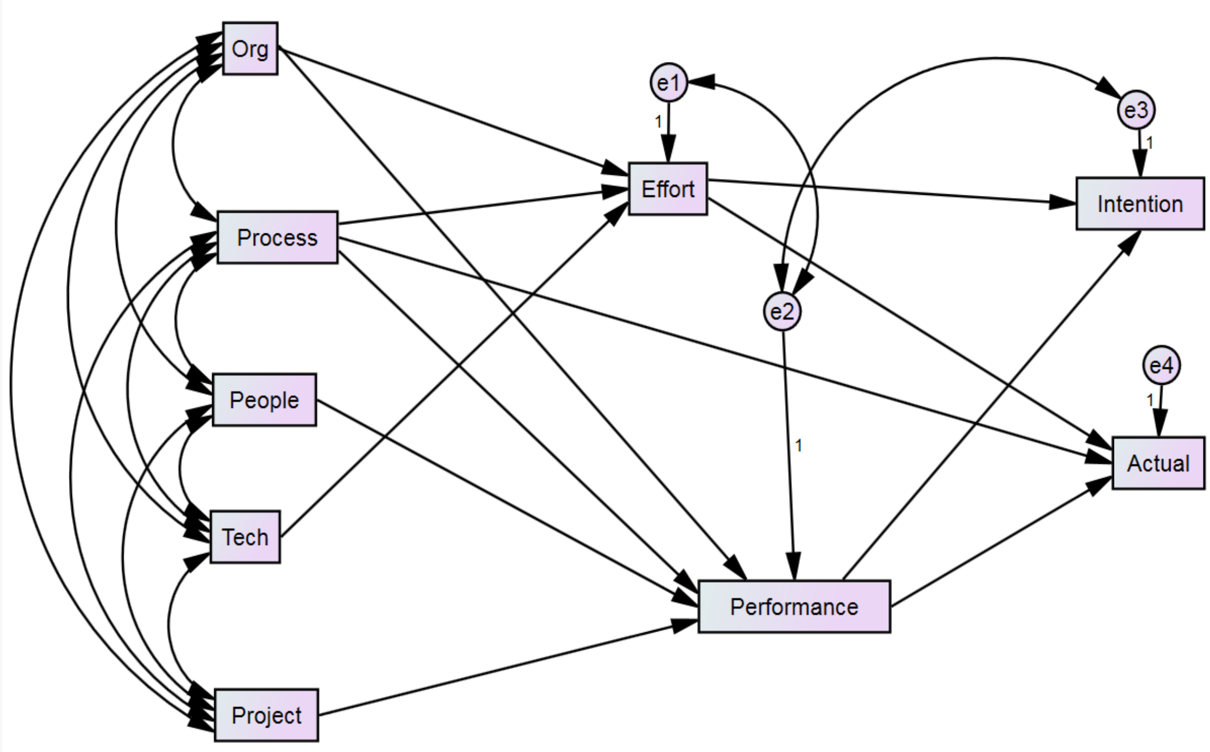


Figure 7.1: Model of agile software development project in South Africa

7.5 Summary

Having explained the results attained from the research, reviewed the research questions, considered the hypotheses, and explored the implications of the results in relation to the hypothesised causal effects between the decisions variables, this chapter has concluded by provided an agile software development model that is general enough to be used in software projects in South Africa.

CHAPTER 8: EVALUATION OF THE RESEARCH

8.1 Introduction

This chapter evaluates the research study, describing the importance of the research and the methods that were carried out to attain the study outcomes. The chapter describes the contribution this research study makes to the IT and software engineering body of knowledge. It also outlines future research opportunities, the importance of the research, an evaluation of the methodologies used, how the research may influence policy in organisations, and limitations of the research study. Recommendations are made regarding policies and measures that companies should follow. The research significance, suitability, methodology, contribution and attainment of the research questions are also explained.

It is important to include such an evaluation since, as Misra et al. (2012) point out, numerous studies in software engineering fall short of significance due to failure to evaluate their research before they publish it.

8.2 Evaluation of the methodological contribution

The evaluation of the contribution to methodology includes testing the research in terms of the method chosen, the theme of the research, the suitability of the topic, the nominated unit of analysis and the data collected for analysis.

8.2.1 Relevance of the used approach

The main aim of this research was to model the critical success factors that affect agile software projects. There are few literature articles written on this area of agile software engineering. The literature findings were combined into a comprehensive model, and the analysis of the primary data was then able to refine the model.

This researcher confirmed that there is much literature on success factors relating to software development. Therefore, the literature review was carried out to identify the relevant critical success factors that have already been recommended. Since the research focused on agile software development projects in South Africa, it was essential to get the views of agile management, software developers and agile experts in this country to determine whether their views on success factors confirm those expressed in the literature. Consequently, open-ended questionnaires and interviews were conducted with carefully selected respondents.

Cross impact analysis was not used because more than 40 critical success factors emerged from the literature. The research used confirmatory factor analysis and exploratory factor analysis to reduce the factors and ensure that factors were valid. Structure equation modelling was used to evaluate the model. This was considered suitable to help explain the influence of the decision variables. Structure equation modelling explains the mediating influence of the latent variables thereby making it a better method to use in this instance than regression analysis.

On reflection, the methods carried out appear to be appropriate and make a contribution to the research knowledge regarding software engineering and agile methodologies.

UTAUT was chosen as most appropriate framework for this research study after an assessment of numerous other theories was implemented (including TRA, TPB, and TAM) and their drawbacks underlined and explained. UTAUT includes new decision variables or constructs namely, process construct, project construct, technological construct, people and organisational constructs, which were introduced to replace the facilitating condition and social influence. The moderating factors were revised where experience with technology was adjusted to experience with agile software development projects to specifically fit this research study. Moderating factors were also introduced to the research model, with education used to replace voluntariness of use and to particularly fit the conceptual framework being examined. Furthermore the final model was assessed by using structural equation modelling (SEM) and the effect of the moderating factors examined using means and covariance structural analysis. Multiple group analysis was used, which is a method normally utilised in marketing psychology but is not extensively used in software engineering. Thus, the empirical analysis of the data provides a methodological contribution to the field.

8.2.2 Research theme of the study

This research aimed to discover critical factors influencing agile software development project success. There have been several critical success factors mentioned by different authors in the literation but there has been a lack of a framework which encompasses individual, organisational, effort expectancy, performance expectancy, process and project features, and technology changes. The increase of individual factors and technology changes means that there is a need to better understand factors that influence agile projects in companies or organisation in order for the projects to be successful and reduce cost and not waste time. Currently, there is increasing failure of agile software development projects.

In light of the lack of a comprehensive framework and the changing context, determining critical success factors for agile software development projects is relevant to software engineering and agile research study.

8.2.3 Fitness of the topic in the knowledge building process

The purpose of this study is to develop a research model to evaluate the critical success factors that influence the success of agile software development projects informed by agile professionals' views on the success of agile software development projects in South Africa. Obtaining these views was done by means of a questionnaire, which was completed by agile professionals involved in software development and management.

The study was developed around a research model based on the various existing theories, including the TPB, TRA, and UTAUT. The model was then used to determine the critical factors contributing to the success of agile software development through an empirical study.

Therefore, understanding the success factors for agile software development projects and recommending better strategies to address the impacts of these factors is a major step in the knowledge building about software engineering systems.

8.2.4 Relevance of the unit of analysis

As much as agile software development projects are used in companies or organisations, their success or failure often relies on the team members and leaders who practise agile techniques. Therefore, the decision to analyse the views of agile experts, team members and leaders was essential.

8.2.5 Collected data in relation to needed research findings

The research used both secondary and primary data for the study. The approach utilised to gather the secondary data (namely the literature review) was significant because the data was collected from accredited articles such as journals, books and conference proceedings. The articles have been peer reviewed by different agile professionals and software engineering experts and are thus regarded as trustworthy.

The primary data was collected through interviews and questionnaires conducted with agile professionals. The questionnaire was sent to agile professionals in several companies and organisations through email, and interviews were conducted face-to-face with agile managers. This was a cross sectional study across all provinces in South Africa.

Hence, this research study noted that the data analysed was appropriate and suitable to provide the findings.

8.3 Importance of the research

According to Almudarra and Qureshi (2015), software engineering research must not be important to professionals and management only but also to other readers and software practitioners, in terms of both style and content. The importance of the substantive content of the research (as opposed to methodology or analysis techniques) is normally evaluated by the degree to which it engages the reader's curiosity and its potential to create awareness and encourage discourse on current matters (such as the critical success factors of agile software development projects in organisations with relevant information and knowledge).

8.3.1 Applicability of the topic and research output

Numerous researchers, such as Almudarra and Qureshi (2015), admit that the failures of agile software development projects demands for research to create better guidelines for agile management and professionals. The topic of this study is of concern and importance to agile management and professionals because there are limited research studies that have set out to determine the critical success factors of agile software development projects through a structural equation modelling approach. In fact, this research study discovered no literature focusing agile critical success factors using SEM. Agile approaches to software development are becoming increasingly popular and have a big market share in software engineering due to the results obtained from this approach.

Therefore, this research study has provided an important contribution through the development of an agile software critical success factor framework.

8.3.2 Contribution to discourse on current matters

Most software development organisations depend on either traditional or agile methodologies. The need to integrate all activities within critical success factors of the organisation have led to the development of this research model.

Since the adoption of agile methodologies is relatively new (within the last eight years) in South Africa where this study research was carried out, the agile critical success factors model developed is timeous and relevant.

8.4 Research contribution of the study

This study explained the critical success factors affecting agile software development projects in terms of the primary objective of the research study. The research sought to develop and validate a model that could function as a guideline for agile software development professionals.

The results of this study have led to the development of a comprehensive model that has been developed for software engineering development companies to provide guidelines for agile management and agile professionals.

8.4.1 Contributions to agile software development projects

The model development was carried out through use of literature and theoretical frameworks that were reviewed, as well as interviews done as part of the research study. The conceptual effect of this is that it shows that UTAUT is a suitable underpinning reference model to use when exploring of use of agile methodologies. UTAUT was developed in and has often been tested on data from developed countries. This research proves that it is also suitable for use in a developing world context as the agile critical success factors model has been based on data from South Africa, a developing country, and the results demonstrated comparable features to those of studies done in developed countries.

Moreover, several research studies that examine agile software development projects critical success factors have relied on the literature merely. This study, however, used interviews and the literature review to identify and analyse the critical success factors, through use of confirmatory factor analysis and structural equation modelling.

In order to identify the critical success factors that influence agile software development projects, a number of research studies have relied on factors in the literature from acceptance and use in other areas of IT. However, this approach has not been enough, thus the researcher interviewed respondents of the agile allegiance group (agile professionals) as well.

The same or similar critical success factors are frequently referred to by more than one term, and researchers use terms interchangeably. This research used different techniques to remove repetitions of questions. The content analysis was carried out to assess the factors and to remove duplications which mighty occur through use of different names that refer to the same success factor. The decision was taken to approve the significance and prominence of success factors in agile software development projects. Interview questions were used to discover the main factors of agile software development projects and, finally, confirmatory factor analysis and structural equation modelling were used to authenticate the model.

The model thus used qualitative and quantitative research methodologies to improve the validity and reliability of the research results. The mixed methods approaches has been supported by several researchers but has rarely been carried out in software engineering research. This utilisation of both quantitative and qualitative approaches provides a contribution towards the methodologies carried out in software engineering studies.

8.4.2 Contributions to agile management, software developers and agile experts

Essentially, this study examined and recognised factors that need to be addressed by agile management, agile experts and software developers. The factors consist of performance expectancy factors, effort expectancy factors, use and intention factors, actual success factors, project factors, organisational factors, people factors, project factors and technological factors.

The study confirmed that performance expectancy factors and effort expectancy factors are key antecedents to intention factors which have a causal relationship with actual success factors. Agile experts, agile management and software developers must make sure that team members find it easy to use agile methodologies and believe that this will improve performance in order to promote success.

The empirical results of the study provide new insights into the success factors influencing agile management, team members and experts with regards to the agile software development, an area of research which is still in its infancy. The research showed that agile professionals believe that agile methodologies enable them to accomplish tasks more efficiently, usefully, and productively, and that it increases the chances of getting a promotion. This positive performance expectancy is influential in the delivery of a good product or software projects that meets all requirements and objectives, delivers software projects on time and delivers software projects within the estimated effort and cost.

In terms of the study, it appears that organisational factors are the key critical success factors, since these were the main contributing construct which enhanced performance expectancy factors. Organisations and companies can help to avoid projects failure by implementing agile practices discussed in chapter 2 which will also lead to a reduction in costs and time of completion of projects and which result in quality and successful projects. A well-documented policy framework and an implementation guide should be drawn up by these organisations to reduce project failure through a focus on critical success factors.

8.4.3 Methodology contribution

The extension of UTAUT in this study and the validation of the developed theoretical framework provided a new approach to identifying and confirming the critical success factors of agile methodologies in software development. In addition, the determination of the main factors of agile methodologies through literature and interviews using thematic analysis has created a way forward in the study. Several studies have only listed factors and called them main or critical success factors, leaving an unanswered question of how and why they are the main factors (Chow & Cao, 2008; Ramesh et al., 2010). This study provided a systematic methodological approach to resolve the criticism that is usually made about studies that set out to determine critical factors. Furthermore, the methodological approach in this study provided a base for empirical confirmatory analysis that could be used by organisations implementing agile methodologies. This methodology is a key contribution to the critical success factors body of knowledge.

8.4.4 Theoretically

This study examined the critical factors that affect the use of agile methodologies, and recognised the degree of influence these factors have on software development projects. The research adapted UTAUT by introducing new decision variables and moderating factors, thereby formulating a new theoretical framework. The researcher proposed a model which is formulated from numerous previous theories and models, seeking to broaden the view of critical success factors of software projects using agile methodologies. Therefore, this contributes to the literature on software projects generally, and agile methodologies in particular.

8.4.5 Practically

Companies spend lots of money acquiring necessary skills in software development which are often underutilised. This research provides an insight into the critical success factors for successful use of agile methodologies. When informed of these critical success factors and their impacts on practice, management could leverage this information to develop better approaches that lead to better use of agile methodologies in software projects. The theoretical framework developed by this study acts as a standard and guideline for appropriate use of agile methodologies and of the critical success factors within organisations. Therefore, agile managers will be able to rely on informed decisions to plan and forecast for the future. This could include which agile methodology to use, the required skills of the programmers and agile managers, and better strategies of dealing with stakeholders and how to construct a more agile appropriate work environment.

8.5 Research impact to policy and organisations

Software development project critical success factors is an area of software engineering that has been widely investigated. Nevertheless, recent research illustrates that failures of agile software development projects are leading the headlines (Bossini & Fernández, 2013). According to Cockburn (2006), the list of research studies on software development projects critical success factors is unending but the failures keep increasing. The increase in agile software development projects failures may be attributed to two simple reasons.

Firstly agile experts, researchers and practitioners have failed to assist agile management to come up with better policies and plans of how to effectively implement agile software development projects. This is caused by failure to inform agile practitioners and agile managements about what to do when a given group of factors becomes a challenge during and after implementation of agile projects. Furthermore, agile management and experts have been focusing on technology rather than on communication and management practices.

Secondly, there has been a lack of in-depth examination of the critical success factors influencing agile software development projects and failure to develop appropriate conceptual frameworks, as well as proper research design and methodologies, and appropriate theoretical grounding.

The study findings show that the future of successful agile software development projects is not focused on the development of technology, but on bridging the roles played by the individuals within companies and across organisational functions. Therefore, the research model in this study can act as an insight into understanding the roles carried out by key actors in agile software development projects.

8.6 Limitations of the research study

1. The research framework does not factor in the personalities and emotions of agile management, and agile experts.

2. The research was done in a developing country context, and data was collected from South Africa sample populations specifically, so does not entirely represent all developing countries. Hence, the results of this study may not be generalizable to all developing countries.

3. The research framework relies on the assumption that humans are rational beings that make systematic judgments.

4. The research framework does not account for unconscious motives.

5. The research does not focus on risk and failure factors such as the influence of a company's broader management and non-technical team members.
6. There might be differences between the technologies used among the study population of the provinces of South Africa or among the different countries.
7. The data used in this research study was collected at one point in time (cross-sectional survey only).
8. Further data is required in future to compare the critical success factors in the provinces of South Africa.
9. The study showed a need to carry out research using longitudinal studies. Future studies could focus on investigating how agile management and experts' perceptions of critical success factors vary over time. This will help explain agile project success trends.
10. Due to resource constraints the research was restricted to South Africa and no follow up was done with participants.

8.7 Recommendations

In terms of the findings from this research study, the following recommendations are made:

1. Agile management and experts need to use agile methodologies in software development projects based on agile policy guidelines. They must focus on the most important critical success factors, such as effort expectancy factors, in order to understand the behaviour and belief of agile managers and programmers, and tailor agile methodologies accordingly and in line with agile practices.

2. This study provides researchers with important knowledge regarding the practice of agile practitioners which can be used for further research and for encouraging agile experts and programmers to employ agile resources more efficiently and effectively in a sustainable way.

3. It is recommended that companies implement the principles in the agile manifesto and that this document should be available to all.

4. Corporate sponsors should reward team members who implement agile methods with appropriate incentives to further improve their efficiency and effectiveness.

5. An agile expert or programmer's intention and actual success is highly dependent on the influence of *organisation factors towards performance expectancy factors*, *performance expectancy factors towards intention behaviour factors*, *organisation factors towards effort expectancy factors* and *effort expectancy factors towards actual success factors*.

   Companies must therefore educate new staff and management on agile practices from the first day so that all staff engage appropriately with agile practices right from the beginning.

   One way of doing this would be for the universities to introduce subjects on agile methodologies into the university curriculum in collaboration with information technology companies and software experts.

8.8 Future research of the study

In view of the findings of this research, the following recommendations are made for future research:

1. A comparative study across the nine provinces of South Africa could be interesting, as South Africa is a diverse country and each province might prove to have different technology patterns influenced by different socio-economic and political circumstances.
2. Future studies could assess agile software development projects by applying longitudinal surveys rather than a slice-time method as used in this research. Using data collected over a longer time period will help researchers to forecast possible trends in agile project success in companies or organisations.
3. Future investigations could be carried out to find out whether the organisational, performance expectancy and effort expectancy factors continue to be the most significant predictors of effort expectancy, intention behaviour and actual success factors for agile. It is not clear whether the findings of the current research study are distinctive or can be generalised to other contexts.
4. Future research must consider using a post-intentional approach. In this approach, respondents are not just asked about what they intend to do, but are also asked to articulate a very specific plan about how they could go about attaining their goal. In this way, an individual is forced to think about the realities of their plan, which could otherwise be too ambiguous. An implementation post-intention approach might better be able to predict the behaviour of agile practitioners and the success of agile software development projects.
5. Future investigations must examine the critical risks and challenges which agile software development projects face.
6. This research study discovered that the level of education is significant when it comes to agile software development projects. More research must be done into how education affects agile software development projects.

8.9 Conclusion

This research examined the perceptions of agile managers and experts regarding critical success factors for agile projects, through the use of the study model. It was found that performance expectancy, process and effort expectancy factors have a causal relationship with actual success factors, and that effort expectancy factors and performance expectancy factors have casual effects on intention behaviour factors. Furthermore, organisational factors, process factors, people factors, technological factors and project factors have a direct effect on the effort expectancy factors of those using agile methodologies. The structural equation modelling shows that organisational factors were are the dominant factor influencing performance expectancy factors. These findings will assist agile management, experts and researchers to concentrate on the most significant components so that the most critical success factors can be better understood and applied in the organisation.

To encourage agile management and experts to translate their effort expectancy into actual project success, agile guidelines and policies must be designed and implemented tailored to individual companies. Also companies must design an environment that enables agile experts and management to embrace an agile approach. Organisational factors regarding agile software development success were found to be statistically significant but performance expectancy is the single-most important factor which needs to be taken into consideration when promoting agile software projects in South Africa (as shown from Table 5.19).

The research considered various potential moderating factors as well. There was a significant difference between the level of education and the paths of the research model, meaning that level of education as an external factor is a major contributing factor, particularly with regards to the level *process factors towards effort expectancy factors*. There was no significant difference between *gender and decision variables*, *level of experience and decision variables* and *age and decision variables*. This means that age, gender and level of experience do not affect the decision variables in terms of practising agile methodologies.

The research questions were reliable and valid as measured by composite reliability, Cronbach's alpha, exploratory factor analysis (EFA) and confirmatory factor analysis (CFA). Gender, level of experience, age and level of education were also almost equally distributed.

In conclusion, the hypotheses, objectives and research questions were answered and the findings of this study may help to promote greater success in agile projects, improving quality, delivering to scope, promoting timeliness and reducing cost of agile software development projects by encouraging efficient and effective use of agile guidelines.

REFERENCE

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). Agile software development methods review and analysis. *VTT Publications, 478*(478), 167-168.

Acharya, A. S., Prakash, A., Saxena, P., & Nigam, A. (2013). Sampling: Why and how of it. *Indian Journal of Medical Specialties, 4*(2), 330-333.

Adams, D. A., Nelson, R. R., & Todd, P. A. (1992). Perceived usefulness, ease of use, and usage of information technology: a replication. *Management Information Systems Quarterly*, 227-247.

Agarwal, R., & Prasad, J. (1998). A conceptual and operational definition of personal innovativeness in the domain of information technology. *Information systems research, 9*(2), 204-215.

Agarwal, R., & Prasad, J. (2000). A field study of the adoption of software process innovations by information systems professionals. *IEEE Transactions on Engineering Management, 47(3)*, 295-308.

Ahmed, A., Ahmad, S., Ehsan, N., Mirza, E., & Sarwar, S. (2010). *Agile software development: Impact on productivity and quality.* Paper presented at the Management of Innovation and Technology (ICMIT), 2010 IEEE International Conference on.

Ajzen, I. (1985). From intentions to actions: a theory of planned behaviour. *In J. Kuhl & J. Beckmann (Eds.), Action control: From cognition to behaviour (pp. 11- 39).* Berlin, Heidelber, New York: Springer-Verlag.

Ajzen, I. (1991). The theory of planned behaviour. *Organisational Behaviour and Human Decision Processes, 50*(2), 179-211.

Ajzen, I., & Fishbein, M. (1975). *Belief, attitude, intention and behaviour: an introduction to theory and research*. Reading, MA: Addison-Wesley.

Ajzen, I., & Fishbein, M. (1977). Attitude-behaviour relations: a theoretical analysis and review of empirical research. *Psychological bulletin, 84*(5), 888.

Ajzen, I., & Fishbein, M. (1980). *Understanding attitudes and predicting social behaviour*. Englewood Cliffs, NJ: Prentice Hall.

Ajzen, I., & Fishbein, M. (2000). Attitudes and the attitude-behaviour relation: reasoned and automatic processes. *European review of social psychology, 11*(1), 1-33.

Ajzen, I., & Fishbein, M. (2004). Questions raised by a reasoned action approach: Comment on Ogden (2003).

Al Tamimi, A. (2014). Empirical investigation on success factors in adapting agile methodology in software development at public organisations. M.S.c. dissertation, Dubai.

Almahamid, S. M. (2013). E-government system acceptance and organisational agility: Theoretical framework and research agendas. *Business and Management, 5*(1).

Almudarra, F., & Qureshi, B. (2015). Issues in adopting agile development principles for mobile cloud computing applications. *Procedia - Procedia Computer Science, 52*, 1133-1140.

Ambler, S. (2009). The agile scaling model (ASM): adapting agile methods for complex environments. *Environments*(December), 1-35.

Ambler, S. W. (2006). The agile edge-unifled-and agile-from objectory to RUP to EUP, the unified process has continued to evolve. now, with its marriage to agility, we have a simplified instantiation tailored for. *Software Development, 14*(1), 49-51.

Asnawi, A. L., Gravell, A. M., & Wills, G. B. (2012). *Emergence of agile methods: Perceptions from software practitioners in Malaysia.* Paper presented at the 2012 Agile India.

Augustine, S., Payne, B., Sencindiver, F., & Woodcock, S. (2005). Agile project management: Steering from the edges. *Communications of the ACM, 48*(12), 85-89.

Babbie, E.R (2015). *Observing ourselves: Essays in social research*: Waveland Press.

Babbie, E., Wagner III, W. E., & Zaino, J. (2015). *Adventures in social research: Data analysis using IBM® SPSS® statistics*: Sage Publications.

Babbie, E. R. (1990). *Survey research methods* Belmont, CA: Wadsworth

Bagozzi, R. P. (1992). The self-regulation of attitudes, intentions and behaviour. *Social Psychology Quarterly*, 178-204.

Bagozzi, R. P., & Edwards, J. R. (1998). A general approach for representing constructs in organisational research. *Organisational Research Methods, 1*(1), 45-87.

Bahli, B., Benslimanne, Y., & Yang, Z. (2011). *The impact of absorptive capacity on the ex-post adoption of agile methods: The case of extreme programming model.* Paper presented at the Industrial Engineering and Engineering Management (IEEM), 2011 IEEE International Conference on, (pp. 1660-1664). IEEE, 2011.

Balnaves, M., & Caputi, P. (2001). *Introduction to quantitative research methods: An investigative approach*: Sage.

Bandura, A. (1977). Self-efficacy: toward a unifying theory of behavioural change. *Psychological Review, 84*(2), 191.

Bandura, A. (1986). *Social foundations of thought and action*. Englewood Cliffs, NJ.

Barlett, J. E., Kotrlik, J. W., & Higgins, C. C. (2001). Organisational research: Determining appropriate sample size in survey research. *Information Technology, Learning, and Performance Journal, 19*(1), 43.

Barrett, P. (2007). Structural equation modelling : Adjudging model fit. *Personality and Individual Difference, 42*(5), 815-824.

Bavelas, J. B. (1995). Quantitative versus qualitative. *Social Approaches to Communication*, 49-62.

Beck, K. (1999). Embracing change with extreme programming. *Computer, 32*(10), 70-77.

Benbasat, I., & Barki, H. (2007). Quo vadis TAM?. Journal of the association for information systems, 8(4), 7.

Bishop, D., & Deokar, A. (2014). *Toward an understanding of preference for agile software development methods from a personality theory perspective.* Paper presented at the 2014 47th Hawaii International Conference on System Sciences.

Boehm, B. (2002). Get ready for agile methods, with care. *Computer, 35*(1), 64-69.

Boehm, B., & Turner, R. (2003a). *Balancing agility and discipline: A guide for the perplexed*: Addison-Wesley Professional.

Boehm, B., & Turner, R. (2003b). Observations on balancing discipline and agility. In *Agile Development Conference, 2003. Proceedings of the* (32-39). IEEE.

Boehm, B., & Turner, R. (2003c). People factors in software management: Lessons from comparing agile and plan-driven methods. *CrossTalk: The Journal of Defense Software Engineering, 16*(12), 4-8.

Boehm, B., & Turner, R. (2003d). Using risk to balance agile and plan-driven methods. *Computer, 36*(6), 57-66.

Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organisations. *Software, IEEE, 22*(5), 30-39.

Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer, 21*(5), 61-72.

Boehm, B. W., Brown, J. R., & Lipow, M. (1976). Quantitative evaluation of software quality. In *Proceedings of the 2nd international conference on Software engineering* (592-605). IEEE Computer Society Press.

Boehm, B. W., & Ross, R. (1989). Theory-W software project management principles and examples. *Software Engineering, IEEE Transactions on, 15*(7), 902-916.

Bossini, J., & Fernández, A. R. (2013). Using agile methodologies in people management. *RPM, 10*(1), 33-42.

Bryman, A. (2015). *Social research methods*: Oxford university press.

Bullen, C. V., & Rockhart, J. F. (1981). *A primer on critical success factors.* Massachusetts Institute of Technology, Sloan School of Management, Center for Information Systems Research: Cambridge, Massachusetts.

Byrne, B. M. (2001). Structural equation modeling with AMOS, EQS and LISREL: Comparative approaches to testing for the factorial validity of a measuring instrumennt. *International Journal of Testing, 1*(1), 55-86.

Byrne, B. M. (2013). *Structural equation modeling with AMOS: Basic concepts, applications, and programming*. Routledge.

Byrne, B. M., & Stewart, S. M. (2006). Teacher's corner : The MACS approach to testing for multigroup invariance of a second- order structure : A walk through the process. . *Structural Equation Modeling, 13*(2), 287-321.

Byrne, B. M., Stewart, S. M., Kennard, B. D., & Lee, P. W. H. (2007). The beck depression inventory-II : Testing for measurement equivalence and factor mean differences across Hong Kong and American adolescents. *International Journal of Testing, 7*(3), 293-309.

Cabrera, A., Collins, W. C., & Salgado, J. F. (2006). Determinants of individual engagement in knowledge sharing. *The International Journal of Human Resource Management*, *17*(2), 245-264.

Carey, M. A., & Asbury, J. E. (2016). *Focus group research*. Routledge.

Ceschi, M., Silitti, A., Succi, G., & De Panfilis, S. (2005). Project management in plan-based and agile companies. *IEEE software, 22*(3), 21-27.

Chaleunvong, K. (2009). Data collection techniques. *Training Course in Reproductive Health Research Vientine. Available on: http://www. gfmer. ch/Activites_internationales_Fr/Laos/Data_collection_tecniques_Chaleunvong_Laos _2009. htm access on October 20th*.

Chan, F. K., & Thong, J. Y. (2009). Acceptance of agile methodologies: A critical review and conceptual framework. *Decision Support Systems*, *46*(4), 803-814.

Chen, F. F., Sousa, K. H., & West, S. G. (2005). Teacher's corner : Testing measurement invariance of second-order factor models. *Structural Equation Modeling, 13*(2), 287-321.

Cheung, G. W., & Rensvold, R. B. (2002). Evaluating goodness-of-fit Indexes for testing measurement invariance *Structural Equation Modeling: A Multidisciplinary Journal, 9*(2), 233-255. doi: 10.1207/S15328007SEM0902

Chiyangwa, T. B., & Alexander, P. T. (2016). Rapidly co-evolving technology adoption and diffusion models. *Telematics and Informatics*, *33*(1), 56-76.

Choi, Y. S. (2000). An empirical study of factors affecting successful implementation of knowledge management. PhD. Thesis, University of Nebraska - Lincoln.

Chow, T., & Cao, D. B. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software, 81*(6), 961-971.

Cockburn, A. (2002). Agile software development joins the" would-be" crowd. *Cutter IT Journal, 15*(1), 6-12.

Cockburn, A. (2006). *Agile software development: the cooperative game*. Pearson Education.

Cockburn, A., & Highsmith, J. (2001). Agile software development: The people factor. *Computer, 34*(11), 131-133.

Cody-Allen, E., & Kishore, R. (2006). An extension of the UTAUT model with e- eqality, trust & satisfaction constructs. *CPR, 2006*, 183-189.

Cohen, D., Lindvall, M., & Costa, P. (2004). An introduction to agile methods. *Advances in Computers*, *62*, 1-66.

Cohen, J., Cohen, P., West, S. G., & Aiken, L. S. (2013). *Applied multiple regression/correlation analysis for the behavioural sciences*. Routledge.

Cohn, M. (2009). *Succeeding with agile: Software development using scrum*. New York City.

Cohn, M., & Ford, D. (2003). Introducing an agile process to an organisation. *Computer, 36*(6), 74-78.

Compeau, D. R., & Higgins, C. A. (1995a). Application of social cognitive theory to training for computer skills. *Information Systems Research, 6*(2), 118-143.

Compeau, D. R., & Higgins, C. A. (1995b). Computer self-efficacy: development of a measure and initial test. *Management Information Systems Quarterly*, 189-211.

Cordeiro, L., Mar, C., Valentin, E., Cruz, F., Patrick, D., Barreto, R., & Lucena, V. (2008). An agile development methodology applied to embedded control software under stringent hardware constraints. *ACM SIGSOFT Software Engineering Notes, 33*(1), 5-5.

Creswell, J. W. (2013). *Research design: Qualitative, quantitative, and mixed methods approaches*. London. Sage publications.

Creswell, J. W. (2014). *A concise introduction to mixed methods research*. Sage Publications.

Creswell, J. W., & Clark, V. L. P. (2007). *Designing and conducting mixed methods research*. University of Nebraska-Lincoln

Creswell, J. W., Hanson, W. E., Plano, V. L. C., & Morales, A. (2007). Qualitative research designs selection and implementation. *The Counseling Psychologist, 35*(2), 236-264.

Curtis, M. B., & Payne, E. A. (2008). An examination of contextual factors and individual characteristics affecting technology implementation decisions in auditing. *International Journal of Accounting Information Systems, 9*(2), 104-121.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *Management Information Systems Quarterly*, 319-340.

Davis, F. D. (1993). User acceptance of information technology: System characteristics, user perceptions and behavioural impacts. *International Journal of Man-machine Studies, 38*(3), 475-487.

Davis, F. D., Bagozzi, R. P., & Warshaw, P. R. (1989). User acceptance of computer technology: A comparison of two theoretical models. *Management Science, 35*(8), 982-1003.

Davis, F. D., Bagozzi, R. P., & Warshaw, P. R. (1992). Extrinsic and intrinsic motivation to use computers in the workplace. *Journal of Applied Social Psychology, 22*(14), 1111-1132.

De Vaus, D. A., & de Vaus, D. (2001). *Research design in social research*. London: Sage.

De Villiers, M. R. (2012). Models for interpretive information systems research, part 2: Design research, development research, design - science research and design - based research - a meta-study and examples. In M. Mora, O. Gelman, A. Steenkamp & M. S. Raisinghani, (eds). *Research Methodologies, Innovations and Philosophies in Software Systems Engineering and Information Systems*.

DeLone, W. H., & McLean, E. R. (2002). Information systems success revisited. In *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on* (pp. 2966-2976). IEEE.

Dion, P. A. (2008). Interpreting structural equation modeling results: a reply to Martin and Cullen. *Journal of Business Ethics, 83*(3), 365-368.

Du Plooy, N. F. (1998). *An analysis of the human environment for the adoption and use of information technology*. PHD Unpublished Thesis, University of Pretoria.

Dyba, T., & Dingsoyr, T. (2009). What do we know about agile software development? *IEEE software, 26*(5), 6-9.

Dyba, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and software technology, 50*(9-10), 833-859.

Dyba, T., & Dingsøyr, T. (2015). *Agile project management: from self-managing teams to large-scale development.* Paper presented at the 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering.

Ecuyer, A. L., & Ahmed, S. A. (2016). Controlling change on agile software development projects. *4*(1), 42-49.

Eriksson, P., & Kovalainen, A. (2015). *Qualitative methods in business Research: A practical guide to social research*. Sage.

Fabrigar, L. R., Porter, R. D., & Norris, M. E. (2010). Some things you should know about structural equation modeling but never thought to ask. *Journal of Consumer Psychology, 20*(2), 221-225.

Farhan, S., Tauseef, H., & Fahiem, M. A. (2009, 2009). *Adding agility to architecture tradeoff analysis method for mapping on crystal*.

Felsing, J. M., & Palmer, S. R. (2002). A practical guide to feature-driven development. *IEEE software, 7*, 67-72.

Fink, A. (2015). *How to conduct surveys: A step-by-step guide*. Sage Publications.

Fishbein, M., & Ajzen, I. (1975). *Belief, attitude, intention and behaviour: An introduction to theory and research*.

Fishbein, M., & Ajzen, I. (2005). Theory-based behaviour change interventions: comments on Hobbis and Sutton. *Journal of Health Psychology, 10*(1), 27-31.

Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Software Development, 9*(8), 28-35.

French, B. F., & Finch, W. H. (2006). Confirmatory factor analytic procedures for the determination of measurement invariance confirmatory *Structural Equation Modeling, 13*(3), 378-402.

Gioia, D. A., Corley, K. G., & Hamilton, A. L. (2013). Seeking qualitative rigor in inductive research notes on the Gioia methodology. *Organisational Research Methods, 16*(1), 15-31.

Glass, R. L. (2002). *Facts and fallacies of software engineering*. Addison-Wesley Professional.

Goh, J. C.L., Pan, S. L., & Zuo, M. (2013). Developing the agile IS development practices in large-scale IT projects: The trust-mediated organisational controls and IT project team capabilities perspectives. *Journal of the Association for Information Systems, 14*(12), 722-722.

Golafshani, N. (2003). Understanding reliability and validity in qualitative research. *The Qualitative Report, 8*(4), 597-607.

Golob, T. F. (2003). Structural equation modeling for travel behaviour research. *Transportation Research Part B: Methodological, 37*, 1-25.

Goodhue, D. L. (1995). Understanding user evaluations of information systems. *Management science, 41*(12), 1827-1844.

Goodhue, D. L. (2007). Comment on Benbasat and Barki's "Quo Vadis TAM" article. *Journal of the Association of information Systems, 8*(4), 219-224.

Goodhue, D. L., & Thompson, R. L. (1995). Task-technology fit and individual performance. *Management Information Systems Quarterly*, 213-236.

Greene, K., Hale, J. L., & Rubin, D. L. (1997). A test of the theory of reasoned action in the context of condom use and AIDS. *Communication Reports, 10*(1), 21-33.

Grinnell Jr, R. M., & Unrau, Y. (2005). *Social work research and evaluation: Quantitative and qualitative approaches*. Cengage Learning.

Hair, J., Bush, R. P. & Ortinau, D. J. (2000). *Marketing research: A practical approach for the new millennium.* Burr Ridge: Irwin, R D.

Hair, J. F., Black, B., Anderson, R., & Tatham, R. (1995). Multivariate Data Analysis: Text and readings.

Hale, J. L., Householder, B. J., & Greene, K. L. (2002). The theory of reasoned action. *The Persuasion Handbook: Developments in Theory and Practice*, 259-286.

Hall, B., & Howard, K. (2008). A synergistic approach conducting mixed methods research with typological and systemic design considerations. *Journal of mixed methods research*, *2*(3), 248-269.

Hansson, C., Dittrich, Y., & Zarnak, S. (2006). How agile are industrial software development practices ? *, 79*, 1295-1311.

Hartung, J., & Knapp, G. (2005). Multivariate multiple regression. *Wiley StatsRef: Statistics Reference Online*.

Hardgrave, B. C., Davis, F. D., & Riemenschneider, C. K. (2003). Investigating determinants of software developers' intentions to follow methodologies. *Journal of Management Information Systems*, *20*(1), 123-151.

Hayes, N. (2000a). *Doing psychological research*. Taylor & Francis Group.

Hayes, N. (2000b). *Doing psychological research: Gathering and analysing data*. Open University Press.

Hazzan, O., & Dubinsky, Y. (2010). *Students' cooperation in teamwork: Binding the individual and the team interests.* Paper presented at the Proceedings of the ACM international conference companion on Object Oriented Programming Systems Languages and Applications Companion.

Henning, E., Van Rensburg, W., & Smit, B. (2004). *Finding your way in qualitative research*. Van Schaik Pretoria.

Highsmith, J. (2002a). *Agile software development ecosystems*. Addison-Wesley Longman Publishing Co., Inc.

Highsmith, J. (2002b). What is agile software development ? (October).

Highsmith, J., & Cockburn, A. (2001). Agile software development. *The business of Innovation Computer, 34*(9), 120-127.

Hu, L. T., & Bentler, P. M. (1999). Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives. *Structural Equation Modeling: A Multidisciplinary Journal, 6*(1), 1-55.

Huisman, M. (Nd). The influence of organisational culture on the use of systems development methodologies. *Management Information Systems Quarterly*, 35-58.

Humphrey, W. S. (1989). Managing the software process. *Addison-Wesley Professional. Humphrey, WS, & Curtis, B.(1991). Comments on a critical look'[software capability evaluations]. Software, IEEE, 8*(4), 42-46.

Iacobucci, D. (2010). Structural equations modeling : Fit indices, sample size and advanced topics. *Journal of Consumer Psychology, 20*(1), 90-98.

Iivari, J., & Huisman, M. (2007). The relationship between organisational culture and the deployment of systems development methodologies. *MIS Quarterly*, 35-58.

Iivari, J., & Iivari, N. (2011). The relationship between organisational culture and the deployment of agile methods. *Information and software technology, 53*(5), 509-520.

Industrial Development Corporation. (2012). Industrial Development Corporation predicts double-digit growth of IT spending by South African Government [Online], Available: http://www.idc.com/getdoc.jsp?containerId=prZA23758612 [Accessed 25 December 2014]

Johansson, C. (2000). Surprising Results from a Measurement Study-is Software Measurement an Exaggerated and Over-emphasised Area?. *Communicating, Measuring and Preserving Knowledge in Software Development*, 119.

Johnson, R. B., Onwuegbuzie, A. J., & Turner, L. A. (2007). Toward a definition of mixed methods research. *Journal of Mixed Methods Research, 1*(2), 112-133.

Johnson, R. B., Onwuegbuzie, A. J., & Turner, L. A. (2016). Mixed methods research: A research paradigm whose time has come. *Educational researcher*, *33*(7), 112-133.

Joseph, N. (2013). A predictive model for information technology project success. PhD thesis, University of Johannesburg, Johannesburg.

Joseph, N., Marnewick, C., & Santana, M. J. (2016). *Agile software development and IT project performance in South Africa: a positive relationship*?. Paper presented at the proceedings on International Association for Management of Technology.

Jun, L., Qiuzhen,W., & Qingguo, M. (2011). The Effects of Project Uncertainty and Risk

Management on IS Development project Performance: A Vendor Perspective. *International Journal of Project Management*, *2*9, 923-933.

Kalogiratou, Z., Monovasilis, T., Ramos, H., & Simos, T. E. (2016). Journal of computational and applied a new approach on the construction of trigonometrically fitted two step hybrid methods. *Journal of Computational and Applied Mathematics, 303*, 146-155.

Kaplan, B. B., & Duchon, D. (1988). Combining qualitative and quantitative methods in information systems a case study. (December), 571-587.

Karlstrom, D., & Runeson, P. (2005). Combining agile methods with star-gate project management. *IEEE Software 22*(3), 43-49.

Karlström, D., Runeson, P., & Norden, S. (2005). A minimal test practice framework for emerging software organisations. *Software Testing, Verification and Reliability, 15*(3), 145-166.

Keppel, G. (1991). *Design and analysis: A researcher's handbook*. Prentice-Hall, Inc.

Klassen, A. C., Creswell, J., Plano Clark, V. L., Smith, K. C., & Meissner, H. I. (2012). Best practices in mixed methods for quality of life research. *Quality of Life Research, 21*(3), 377-380.

Knox, K. (2004). A Researcher's dilemma-philosophical and methodological pluralism. *Electronic Journal of Business Research Methods, 2*(2), 119-128.

Koch, A. S. (2005). *Agile software development: Evaluating the methods for your organisation*. Artech House Boston.

Kolb, B. (2008). *Marketing research: A practical approach*. London: Sage.

Lalsing, V., Kishnah, S., & Pudaruth, S. (2012). People factors in agile software development and project management. *International Journal of Software Engineering & Applications (IJSEA), 3*(1), 117-137.

Larman, C. (2004). *Agile and iterative development: A manager's guide*. Addison-Wesley Professional.

Lee, G., & Xia, W. (2010). Toward Agile: An Integrated Analysis of Quantitative and

Qualitative Field Data on Software Development Agility. *MIS Quarterly, 34*(1), 87-114.

Leedy, P. D., & Ormrod, J. E. (2005). *Practical research*.Saddle River: New Jersey.

Lindvall, M., Basili, V., Boehm, B., Costa, P., Dangle, K., Shull, F., Tesoriero, R., Laurie Williams, L., & Zelkowitz, M. (2002). Empirical findings in agile methods. In *conference on Extreme Programming and Agile Methods* (pp. 197-207). Springer Berlin Heidelberg.

Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., & Kahkonen, T. (2004). Agile software development in large organisations. *Computer*, *37*(12), 26-34.

Lippert, M., Wolf, H., & Roock, S. (2002). *Extreme programming in action: Practical experiences from real world projects*: John Wiley & Sons, Inc.

Livermore, J. A. (2008). Factors that significantly impact the implementation of an agile software development methodology. *Journal of Software, 3*(4), 31-36.

Lucas, J. H. C., & Spitler, C. (1999). Technology use and performance: A field study of broker workstations. *Decision Sciences, 30*(2), 291-311.

Madden, T. J., Ellen, P. S., & Ajzen, I. (1992). A comparison of the theory of planned behaviour and the theory of reasoned action. *Personality and Social Psychology Bulletin, 18*(1), 3-9.

Marshall, C., & Rossman, G. B. (2014). *Designing qualitative research*: Sage publications.

Marnewick, C. (2012). A longitudinal analysis of ICT project success. *Proceedings of the South African Institute for Computer Scientists and Information Technologits Conference.* Pretoria: ACM.

Marnewick, C. & Labuschagne, L. (2009). Factors that influence the outcome of information technology projects in South Africa: An empirical investigation. *Acta Commercii,* 9, 78-89.

Mathieson, K. (1991). Predicting user intentions: comparing the technology acceptance model with the theory of planned behaviour. *Information systems research, 2*(3), 173-191.

Maxwell, J. A. (2012). *Qualitative research design: An interactive approach.* Sage.

McConnell, S. (1996). *Rapid development: Taming wild software schedules*. Pearson Education.

McConnell, S., & Root, D. (1996). Lifecycle planning. *Rapid Development: Taming Wild Software Schedules*. Retrieved March 12, 2015, from http://www.cs.cmu.edu/~aldrich/courses/413/slides/6-planning-2.pdf

McIntosh, C. N. (2007). Rethinking fit assessment in structural equation modelling : A commentary and elaboration on Barrett (2007). *Personality and Individual Difference, 42*(5), 859-867.

Misra, S., Kumar, V., Kumar, U., Fantazy, K., & Akhter, M. (2012). Agile software development practices: Evolution, principles, and criticisms. *International Journal of Quality & Reliability Management, 29*(9), 972-980.

Misra, S. C., Kumar, V., & Kumar, U. (2006). Success Factors of Agile Software Development. In *Software Engineering Research and Practice* (pp. 233-239).

Misra, S. C., Kumar, V., & Kumar, U. (2009). Identifying some important success factors in adopting agile software development practices. *Journal of Systems and Software, 82*(11), 1869-1890.

Moen, K., & Middelthon, A.l. (2015). *Qualitative research methods*: Elsevier Ltd.

Moffett, S., McAdam, R., & Parkinson, S. (2003). An empirical analysis of knowledge management applications. *Journal of Knowledge Management, 7*(3), 6-26.

Montana, P. J., & Charnov, B. H. (2008). Management. New York, Barron's Educational Series. *Inc. Pg*, *333*.

Moore, G. C., & Benbasat, I. (1991). Development of an instrument to measure the perceptions of adopting an information technology innovation. *Information Systems Research, 2*(3), 192-222.

Moore, G. C., & Benbasat, I. (1996). Integrating diffusion of innovations and theory of reasoned action models to predict utilisation of information technology by end-users. *Diffusion and adoption of information technology* (pp. 132-146): Springer.

Morgan, D. (1998). The focus group guidebook (Vol. 1). Thousand Oaks, CA: Sage.

Moustakas, C. (1994). *Phenomenological research methods*. Sage Publications.

Mugenda, O. M. (1999). *Research methods: Quantitative and qualitative approaches*. African Centre for Technology Studies.

Munassar, N. M. A., & Govardhan, A. (2010). A comparison between five models of software engineering. *IJCSI, 5*, 95-101.

Nardi, P. M. (2015). *Doing survey research*. Routledge.

Nasehi, A. (2013). A quantitative study on critical success factors in agile software development projects; case study IT company. M.S.c. Dissertation, University of Boras.

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM, 48*(5), 72-78.

Netemeyer, R., Ryn, M. V., & Ajzen, I. (1991). The theory of planned behaviour. *Organisational Behaviour and Human Decision Processes, 50*(2), 179-211.

Nguyen, D. (2016). Success factors that influence agile software development project success. *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS),* 171-222

Oates, B. J. (2006). *Researching information systems and computing.* London: Sage Publications Ltd.

Olszewska, M., Heidenberg, J., Weijola, M., Mikkonen, K., & Porres, I. (2016). Quantitatively measuring a large-scale agile transformation. *Journal of Systems and Software, 117*, 258-273.

Pallant, J. (2013). *SPSS survival manual*. McGraw-Hill Education (UK).

Parker, S. (2000). From passive to proactive motivation: The importance of flexible role orientations and role breadth self-efficacy. *Applied Psychology*, *49*(3), 447-469.

Parsons, D., Ryu, H., & Lal, R. (2007). The impact of methods and techniques on outcomes from agile software development projects. *Organisational Dynamics of Technology-Based Innovation: Diversifying the Research Agenda* (pp. 235-249): Springer.

Pearl, J. (2000). *Causality: Models, reasoning and inference*. London: Cambridge University Press.

Pichler, R. (2010). *Agile product management with scrum: Creating products that customers love (Adobe Reader)*: Addison-Wesley Professional.

Pillai, K., Phase, A., & Phase, T. (1996). The fountain model and its impact on project schedule. *ACM SIGSOFT Software Engineering Notes, 21*(2), 32-38.

Pomykalski, J. J., Dion, P., & Brock, J. L. (2008). A structural equation model for predicting business student performance. *Journal of Education for Business, 83*(3), 159-164.

Poppendieck, M., & Poppendieck, T. (2003). *Lean software development: An agile toolkit*: Addison-Wesley Professional.

Punch, K. F. (2013). *Introduction to social research: Quantitative and qualitative approaches*. Sage.

Ramesh, B., Cao, L., & Baskerville, R. (2010). Agile requirements engineering practices and challenges: an empirical study. *Information Systems Journal, 20*(5), 449-480.

Reel, J. S. (1999). Critical success factors in software projects. *IEEE software, 16*(3), 18-23.

Reifer, D. J. (2003). XP and the CMM. *IEEE Software, 20*(3), 14-15.

Reifer, D. J., Maurer, F., & Erdogmus, H. (2003). Scaling agile methods. *Software, IEEE, 20*(4), 12-14.

Reinecke, J., Schmidt, P., & Ajzen, I. (1996). Application of the theory of planned behaviour to adolescents' condom use: a panel study. *Journal of Applied Social Psychology, 26*(9), 749-772.

Reiter, B. (2013). The epistemology and methodology of exploratory social science research: Crossing Popper with Marcuse. *The Dialectics of Citizenship: Exploring Privilege, Exclusion, and Racialisation*.

Remenyi, D., & Williams, B. (1998). *Doing research in business and management: An introduction to process and method*. Sage.

Ritchie, J., Lewis, J., Nicholls, C. M., & Ormston, R. (2013). *Qualitative research practice: A guide for social science students and researcher*: Sage.

Rocco, T. S., Bliss, L. A., Gallagher, S., & Pérez-Prado, A. (2003). Taking the next step: Mixed methods research in organisational systems. *Information Technology, Learning, and Performance Journal, 21*(1), 19.

Rodríguez, P., Markkula, J., Oivo, M., & Turula, K. (2012). *Survey on agile and lean usage in finnish software industry.* Paper presented at the proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement.

Rogers, E. M. (1962). *Diffusion of Innovations*. New York: Free Press.

Rogers, E. M. (1983). *Diffusion of innovations*. New York: Free Press.

Rogers, E. M. (2008). *Diffusion of innovations*. New York: Free Press.

Rogers, E. M., & Quinlan, M. M. (2004). Diffusion of Innovations Everett M. Rogers. 1-25.

Rohani, J. M., Yusof, S. M., & Mohamad, I. (2009). The relationship between statistical process control critical success factors and performance : A structural equation modeling approach. *International Conference on Industrial Engineering and Engineering Management*, 1352-1356.

Royce, W. W. (1970). *Managing the development of large software systems.* Paper presented at the proceedings of IEEE WESCON.

Saga, V. L., & Zmud, R. W. (1993). *The nature and determinants of IT acceptance, routinisation, and infusion.* Paper presented at the proceedings of the IFIP TC8 working conference on Diffusion, Transfer and Implementation of Information Technology.

Schatz, B., & Abdelshafi, I. (2005). Primavera gets agile: A successful transition to agile development. *IEEE software*(3), 36-42.

Schreiber, J. B. (2008). Core reporting practices in structural equation modeling. *Research in Social and Administrative Pharmacy, 4*(2), 83-97.

Schwaber, K. (1997). Scrum development process (pp. 117-134): Springer London.

Schwaber, K. (2004). *Agile project management with scrum*: Microsoft press.

Schwaber, K., & Beedle, M. (2002). Agilè software development with scrum. London.

Sharma, M., & Kanekar, A. (2007). Theory of reasoned action & theory of planned behaviour in alcohol and drug education. *Journal of Alcohol and Drug Education*, 51(1), 3.

Sheffield, J., & Lemétayer, J. (2013). Factors associated with the Software Development

Agility of Successful projects. *International Journal of Project Management, 31*(3), 459–472.

Sheppard, B. H., Hartwick, J., & Warshaw, P. R. (1998). The theory of reasoned action: a meta-analysis of past research with recommendations for modifications and future research. *Journal of Consumer Research, 15*(325-343).

Sillitti, A., Ceschi, M., Russo, B., & Succi, G. (2005). *Managing uncertainty in requirements: A survey in documentation-driven and agile companies.* Paper presented at the Software Metrics, 2005. 11th IEEE International Symposium.

Smite, D., Wohlin, C., Gorschek, T., & Feldt, R. (2010). Empirical evidence in global software engineering: a systematic review. Empirical software engineering, 15(1), 91-118.

Sparks, P., Shepherd, R., & Frewer, L. J. (1995). Assessing and structuring attitudes toward the use of gene technology in food production: the role of perceived ethical obligation. *Basic and Applied Social Psychology, 16*(3), 267-285.

Stankovic, D., Nikolic, V., Djordjevic, M., & Cao, D.B.(2013). A survey study of critical success factors in agile software projects in former Yugoslavia IT companies. *The Journal of Systems and Software, 86*(6), 1663-1678.

Stapleton, J. (1997). *DSDM: Dynamic Systems Development Method*. London, England:

Pearson

Steinberg, D. H., & Palmer, D. W. (2003). *Extreme software engineering a hands-on approach*: Prentice-Hall, Inc.

Strode, D. E. (2005). The agile methods: An analytical comparison of five agile methods and an investigation of their target environment. *Unpublished Master of Information Sciences (Information Systems), Massey University, Palmerston North*.

Strode, D. E., Huff, S. L., & Tretiakov, A. (2009). *The impact of organisational culture on agile method use.* Paper presented at the System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on.

Subiyakto, A., & bin Ahlan, A. (2013). *A coherent framework for understanding critical success factors of ICT project environment.* Paper presented at the Research and Innovation in Information Systems (ICRIIS), 2013 International Conference on.

Subramaniam, V., & Hunt, A. (2006). *Practices of an agile developer*: Pragmatic Bookshelf.

Sudhakar, P. (2012). A Model of Critical Success Factors for Software Development. *Journal*

of Entreprise Information Management, 25(6), 537-558.

Sultan, F., & Chan, L. (2000). The adoption of new technology: the case of object-oriented computing in software companies. *IEEE transactions on Engineering Management*, *47*(1), 106-126.

Sutharshan, A. (2011). Enhancing Agile methods for multi-cultural software project teams. *Modern Applied Science, 5*(1), 12-22.

Sutharshan, A. (2013). Human factors and cultural influences in implementing agile philosophy and agility in global software development. PhD. Thesis, Edith Cowan University.

Sutherland, J., Schwaber, K., Scrum, C. C. O., & Sutherl, C. J. (2007). *The scrum papers: Nuts, bolts, and origins of an agile process.* Washington: Newton.

Takeuchi, H., & Nonaka, I. (1986). The new new product development game. *Harvard business review, 64*(1), 137-146.

Taromirad, M., & Ramsin, R. (2008). *Cefam: Comprehensive evaluation framework for agile methodologies.* Paper presented at the Software Engineering Workshop, 2008. SEW'08. 32nd Annual IEEE.

Taylor, S., & Todd, P. (1995a). Assessing IT usage: the role of prior experience. *Management Information Systems Quarterly*, 561-570.

Taylor, S., & Todd, P. A. (1995b). Understanding information technology usage: a test of competing models. *Information Systems Research, 6*(2), 144-176.

Theunissen, W. H. M., Herman, W., & Theunissen, M. (2003). *A case-study based assessment of agile software development.* University of Pretoria.

Thompson, R. L., Higgins, C. A., & Howell, J. M. (1991). Personal computing: toward a conceptual model of utilisation. *MIS quarterly*, 125-143.

Tibenderana, P. K., & Ogao, P. J. (2008). Acceptance and use of electronic library services in Ugandan universities. In Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries (pp. 323-332). ACM.

Tibenderana, P., Ogao, P., Ikoja-Odongo, J., & Wokadala, J. (2010). Measuring levels of end-users' acceptance and use of hybrid library services. *International Journal of Education and Development using Information and Communication Technology, 6*(2), 1F.

Triandis, H. C. (1980a). Reflections on trends in cross-cultural research. *Journal of cross-cultural psychology, 11*(1), 35-58.

Triandis, H. C. (1980b). Values, attitudes and interpersonal behaviour. *In: M.M. Page, Editor, Nebraska Symposium on Motivation, 1979: Beliefs, Attitudes, and Values, University of Nebraska Press, Lincoln (1980)*, 195-259.

Venkatesh, V. (2000). Determinants of perceived ease of use: Integrating control, intrinsic motivation, and emotion into the technology acceptance model. *Information Systems Research, 11*(4), 342-365.

Venkatesh, V., Brown, S. A., & Bala, H. (2013). Bridging the qualitative-quantitative divide: Guidelines for conducting mixed methods research in information systems. *Management Information Systems Quarterly, 37*(1), 21-54.

Venkatesh, V., & Davis, F. D. (2000). A theoretical extension of the technology acceptance model: four longitudinal field studies. *Management Science, 46*(2), 186-204.

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. *Management Information Systems Quarterly*, 425-478.

Wan, J., & Wang, R. (2010). Empirical research on critical Success factors of agile software process improvement. *Journal of Software Engineering and Applications, 3*(12), 11-31.

Warshaw, P. R. (1980). A new model for predicting behavioural intentions: an alternative to Fishbein. *Journal of Marketing Research*, 153-172.

Weilbach, L., & Byrne, E. (2010). A human environmentalist approach to diffusion in ICT policies: a case study of the FOSS policy of the South African government. *Journal of Information, Communication and Ethics*(1), 108-123.

Weinberg, R. S. (1991). Prototyping and the systems development life cycle. *Information System Management, 8*(2), 47-53.

Wetherbe, J. C. (1991). Executive information requirements: Getting it right. *Management Information Systems Quarterly*, 51-65.

Whitworth, E., & Biddle, R. (2007a). Motivation and cohesion in agile teams *Agile processes in software engineering and extreme programming* (pp. 62-69): Springer.

Whitworth, E., & Biddle, R. (2007b). *The social nature of agile teams.* Paper presented at the Agile Conference (AGILE).

Williams, L., Layman, L., & Krebs, W. (2004). Extreme Programming Evaluation Framework for Object-Oriented Languages--Version 1.4. North Carolina State University Department of Computer Science, Raleigh, NC, TR-2004-11.

Williams, M. D., Rana, N. P., Dwivedi, Y. K., & Lal, B. (2011). *Is UTAUT really used or just cited for the sake of it? a systematic review of citations of UTAUT's originating article.* Paper presented at the ECIS.

Williams, R. D. (1984). Management of software development. *Vick, CR and Ramamoorthy, CV (Eds.)*.

Wolff, S. (2012). Scrum goes formal: Agile methods for safety-critical systems*. Paper presented at the Proceedings of the First International Workshop on Formal Methods in Software Engineering: Rigorous and Agile Approaches.*

Yeaman, A. R. (1988). Attitudes, learning and human-computer interaction: an application of the Fishbein and Ajzen model of attitude-behaviour consistency*. In: Proceedings of selected research papers presented at the Annual Meeting of the Association for Educational Communications and Technology* (New Orleans, LA, January 14-19, 1988).

Yew Wong, K. (2005). Critical success factors for implementing knowledge management in small and medium enterprises. *Industrial Management & Data Systems, 105*(3), 261-279.

Yew Wong, K., & Aspinwall, E. (2005). An empirical study of the important factors for knowledge-management adoption in the SME sector. *Journal of knowledge Management, 9*(3), 64-82.

Yu, C.S. (2012). Factors affecting individuals to adopt mobile banking: Empirical evidence from the UTAUT model. *Journal of Electronic Commerce Research, 13*(2), 104.

Zhou, T., Lu, Y., & Wang, B. (2010). Integrating TTF and UTAUT to explain mobile banking user adoption. *Computers in Human Behaviour, 26*(4), 760-767.

APPENDICES

Appendix A: Questionnaire used for the study

Agile Software Development Projects Questionnaire

1. This research is conducted as part of a doctorate program at the University of South Africa.
2. Software is important for the technological advancement of the modern world. Various software engineering methodologies and software development projects have been unsuccessful with regards to software development. This has resulted in malfunctioning in terms of software quality of software projects. In spite of malfunction software projects, the same software projects have high cost of in terms of maintenance and corrective releases of the same software projects to the customers.
3. The purpose of this study is to develop a research model to evaluate the critical success factors that influence the success of agile software development projects by informing agile professionals and about the success of agile software development projects in South Africa, through the use of a survey completed by agile practitioners and software managers.
4. The questionnaire forms part of an investigation that tries to discover the main success factors which affect the actual success of the agile software development projects. Specifically, we are interested in your own experience in the business, behaviour and beliefs regarding agile methodologies. Please read each question carefully and answer it to the best of your ability. There are no correct or incorrect responses; we are merely interested in your personal point of view.
5. All responses to this survey are completely confidential. The instructor of this course will not disclose your responses to the organisation or company staff or society. Please be assured that the information you provide in this study will have no effect on your work.

Please note: 1 represents strongly disagree, 2 represents disagree, 3 represents neutral, 4 represents agree and 5 represents strongly agree on section B for the questionnaire.

SURVEY INSTRUMENT

SECTION A

DEMOGRAPHICAL INFORMATION:

Please provide some basic information regarding the agile projects:

1. What is your age?

16-20	
21-25	
26-30	
31-35	
36-40	
Above 40	

2. Which agile methodology do you used in your organisation or company?

Extreme Programming	
Scrum	
Dynamic Systems Development Method	
Feature-Driven Development	
Adaptive Software Development	
Lean software development	
Crystal	

3. What is the size of the project (number of project team members)?

5 or less team members	
5-10 team members	
10-15 team members	
More than 15 team members	

4. What is the length of the project (in months) which you normally do in your organisation?

10 or less months	
10-20 months	
20-30 months	
More than 30 months	

5. Where is your company location in which you do the project (province)?

Gauteng	
Western Cape	
Mpumalanga	
Eastern Cape	
Northern Cape	
Free State	
Limpopo	
Kwa-Zulu Natal	
Other	

Organisation

Please provide some basic information regarding your organisation:

6. What is your highest level of tertiary education, which is ICT related?

None	
Certificate	
Diploma	
Degree	
Other	

7. What is your company size (ranges of number of employees)?

1-20 employees	
20-40 employees	
40-60 employees	
60-80 employees	
More than 80 employees	

8. How much company revenues (ranges of annual sales dollar amounts) do you have per year?

0-1 000 000	
1 000 000-5 000 000	
5 000 000-10 000 000	
10 000 000-15 000 000	
15 000 000-20 000 000	
20 000 000- 25 000 000	
More than 25 000 000	

9. What is your company's industry (selection of pre-determined industries)?

ICT	
Software Development	
Mobile development and communication	
IT	
Banks	
Manufacturing	
Medical and Auditing	
Financial	
Economic	
Educational	
Aviation	
Other	

10. What is your job responsibility in the project in your organisation?

Project manager,	
Team lead,	
Team member,	
Customer,	
Organisation management,	
Other	

11. What is your level of experience with agile software development projects (in years)?

0-2	
2-4	
4-6	
8-10	
10-12	
Above 12	

12. How many number of agile software development project have been involved with?

0-5	
5-10	
10-15	
15-20	
20-25	
Above 25	

13. What is your gender?

Male	
Female	

SECTION B

SUCCESS FACTORS OF THE AGILE PROJECT

ORGANISATIONAL FACTORS	1	2	3	4	5
1. The project team worked in a facility with proper agile-style work environment, e.g. a dedicated office with pair programming workstations, ample wall spaces for postings, communal area, no separate offices or cubicles, etc.					
2. The project team is collocated, i.e. all team members worked in the same location for ease of communication and casual, and constant contact.					
3. The organisation had a cooperative culture instead of hierarchal.					
4. The organisation had an oral culture placing high value on fluid, face-to-face communication style.					
5. Agile methodology was universally accepted in the organisation.					
6. The organisation had a reward system that was appropriate for agile behaviour.					
7. The project had a committed sponsor or a committed organisation manager. An example of a committed sponsor or manager would be one who would stand up to critics and vouch for the agile method in a non-agile organisational environment.					
8. The project received good management or executive support.					
PEOPLE DIMENSION	**1**	**2**	**3**	**4**	**5**
1. Project management had a good relationship with the customer and is knowledgeable in agile principles and processes.					

	1	2	3	4	5
2. The project team members worked in a cohesive, self-organising teamwork manner, had great motivation and are committed to the project success, i.e. relying on the collective ability of an autonomous team to solve problems and adapt to changing conditions.					
3. Project management had light-touch or adaptive management style, for instance. Flexible working environment, encouraging creative, etc.					
4. The selected project team members had the high technical competence and expertise with regards to problem solving, programming and subject matter.					
PROCESS FACTORS	**1**	**2**	**3**	**4**	**5**
1. The project had strong communication focus and rigorous communication schedule, i.e. face-to-face and instant communication channels (between team members, between the team and management, and between team and customers), daily stand-up meetings, build cycle meetings, etc.					
2. The project followed agile-oriented requirement process, e.g. specifying initial requirements at a very high level, leaving much room for interpretation and adaptation as the project progressed.					
3. The project scope and objectives were well-defined and honoured regular working schedule, i.e. 40-hour work week, no overtime.					
4. The project manager followed an agile-friendly progress tracking mechanism, e.g. using flexible time-boxing or rapid-pace progress measurement techniques instead of document milestones or work breakdown structure.					
5. The project followed agile project management style and had the strong customer commitment and presence, i.e. having at least one customer representative on site working hard and full-time as a member of the project team.					

	1	2	3	4	5
6. The project followed agile-oriented configuration management process, e.g. employing good version control or source code management to accommodate the refactoring efforts and frequent builds.					
7. The customer representative on the project had full authority and knowledge to make decisions on-site, such as approving, disapproving, and prioritising project requirements and changes.					
TECHNOLOGICAL FACTORS	**1**	**2**	**3**	**4**	**5**
1. The project imposed a well-defined coding standard up front, delivered working software regularly within short periods of time and delivered most important features first.					
2. The project pursued simple design, e.g. programmers used the simplest possible design for each module to avoid waste and to facilitate cooperative work.					
3. The project pursued vigorous refactoring activities to ensure the results are optimal and to accommodate well all changes in requirements:					
4. The project maintained the right amount of documentation for agile purpose, i.e. not too focused on producing elaborate documentation as milestones but not ignoring documentation altogether either.					
5. The project followed continuous and rigorous unit and integration testing strategy for each and every iteration.					
6. The project employed proper platforms, technologies, and tools suitable for agility practice, e.g. object-oriented development techniques, tools supporting rapid iterative development, processes supporting refactoring, etc.					
7. The project provided appropriate technical training to the team, including training on subject matter and agile processes.					
PROJECT FACTORS	**1**	**2**	**3**	**4**	**5**
1. The project had a dynamic, accelerated schedule and a small team size (20 members or less).					

2. The project had no multiple, independent teams working together and was of variable scope with emerging requirements.					
3. The project had up-front, detailed cost evaluation completed and approved, and up-front risk analysis completed and assessed for using agile methods.					
4. The project nature was a non-life-critical software project, although it could be business mission-critical software.					
CULTURAL FACTORS	**1**	**2**	**3**	**4**	**5**
1. Our organisation does not have a bureaucratic management structure.					
2. Our organisational culture is customer centric.					
3. Our management has the culture for supporting the decisions of the developers.					
4. Our organisation encourages rapid communication and feedback from customers as culture.					
5. Our organisation has the culture of trusting people.					

SOCIAL FACTORS	1	2	3	4	5
1. Our organisation expert such as developers influence us to agile methodologies.					
2. Our organisation motivates us to use agile methodologies for the success of agile projects.					
3. Our management encourages us to use agile methodologies for the success of agile projects.					
4. Management from different organisation use agile methodologies for the success of agile projects.					
5. Friends from different organisation use agile methodologies for the success of agile projects.					
6. It is expected for me to use agile methodologies for the success of agile projects.					
7. Most people whom I value of my life uses agile methodologies for the success of agile projects.					
8. Most people like me because I use agile methodologies for the success of agile projects.					
ACTUAL SUCCESS OF THE AGILE SOFTWARE DEVELOPMENT PROJECTS	**1**	**2**	**3**	**4**	**5**
1. The project was successful in terms of costs being under budget or within estimates.					
2. The project was successful in terms of timeliness of project completion:					
3. The project was successful in terms of scope of the project being met.					
4. The project was successful in terms of quality of the project outcome or of the resulting software product.					
5. The project was successful in terms of efforts being within estimates finish of the projects.					
6. The project was successful in terms of requirements of the project being met.					
7. The project was successful in terms of the plans intended for the project to accomplish by customer.					
POLITICAL AND NATIONAL LEVEL FACTORS	**1**	**2**	**3**	**4**	**5**

	1	2	3	4	5
1. In my organisation, governmental policies for developing country cause positive influence on the success of software project practices.					
2. My organisation support infrastructure and technological availability in their projects.					
3. External politics influences my organisations and play a major role more especially when it is to do with issues of making decisions. Such as, team member and management selection, hiring of personnel and assignment of tasks etc.					
4. My organisation uses recent hardware and software for the success of agile projects					
INTENTION FACTORS	**1**	**2**	**3**	**4**	**5**
1. I plan to practice agile software development projects on a regular basis.					
2. I will make an effort to practice agile software development projects on a regular basis.					
3. I intend to practice agile software development projects on a regular basis.					
4. I predict to use agile software development projects					
5. Assuming I had agile methodologies knowledge, I intend to use it.					
6. Given that I had access to use agile methodology, I predict that I would use it.					
7. I intend to use agile methodologies on a regular basis for the success of agile projects					
PERFORMANCE EXPECTANCY FACTORS	**1**	**2**	**3**	**4**	**5**
1. I find the agile software development projects useful in my job.					
2. Using the agile software development projects enables me to accomplish tasks more quickly.					
3. Using the agile software development projects increases my productivity at work.					

4. If I use the agile software development projects, I will increase my chances of getting promotion.					
EFFORT EXPECTANCYFACTORS	**1**	**2**	**3**	**4**	**5**
1. My interaction with the agile software projects are clear and understandable.					
2. It would be easy for me to become skillful at using the agile software projects techniques.					
3. I would find the agile software projects easy to use.					
4. Learning to operate the agile software project is easy for me.					
5. It is easy for me to agile methods than traditional methods.					
6. Using agile methods is easy and understandable in my company					
7. Practising to use agile methods is ease to use					

Other:____

Appendix B: Interview questions used for the study

Interview questions

1. What are the social factors that influence agile software development projects?
2. What are the project factors that influence agile software development projects?
3. What are the process factors that influence agile software development projects?
4. What are the political factors that influence agile software development projects?
5. What are the cultural factors that influence agile software development projects?
6. What are the vendor factors that influence agile software development projects?
7. What are the individual or people factors that influence agile software development projects?
8. What are the organisational factors that influence agile software development projects?
9. What are the technological factors that influence agile software development projects?
10. What are the intended success factors that influence agile software development projects?
11. What are the performance expectancy factors that influence agile software development projects?
12. What are the effort expectancy factors that influence agile software development projects?
13. What are the actual success factors that influence agile software development projects?

Appendix C: Letter of approval

UNISA college of science, engineering and technology

Dear Mr Tawanda Blessing Chiyangwa (49918494)

Date: 2015-08-27

Application number: 084/TBC/2015/CSET_SOC

REQUEST FOR ETHICAL CLEARANCE: (An empirical study of success factors in agile software development projects in South Africa)

The College of Science, Engineering and Technology's (CSET) Research and Ethics Committee has considered the relevant parts of the studies relating to the abovementioned research project and research methodology and is pleased to inform you that ethical clearance is granted for your research study as set out in your proposal and application for ethical clearance.

Therefore, involved parties may also consider ethics approval as granted. However, the permission granted must not be misconstrued as constituting an instruction from the CSET Executive or the CSET CRIC that sampled interviewees (if applicable) are compelled to take part in the research project. All interviewees retain their individual right to decide whether to participate or not.

We trust that the research will be undertaken in a manner that is respectful of the rights and integrity of those who volunteer to participate, as stipulated in the UNISA Research Ethics policy. The policy can be found at the following URL:
http://cm.unisa.ac.za/contents/departments/res_policies/docs/ResearchEthicsPolicy_apprvCounc_21Sept07.pdf

Please note that the ethical clearance is granted for the duration of this project and if you subsequently do a follow-up study that requires the use of a different research instrument, you will have to submit an addendum to this application, explaining the purpose of the follow-up study and attach the new instrument along with a comprehensive information document and consent form.

Yours sincerely

Prof Ernest Mnkandla
Chair: College of Science, Engineering and Technology Ethics Sub-Committee

Prof IOG Moche
Executive Dean: College of Science, Engineering and Technology

RECEIVED
2015 -08- 27
OFFICE OF THE EXECUTIVE DEAN
College of Science, Engineering and Technology

University of South Africa
College of Science, Engineering and Technology
The Science Campus
C/o Christiaan de Wet Road and Pioneer Avenue, Florida Park, Roodepoort
Private Bag X6, Florida, 1710
www.unisa.ac.za/cset

UNISA college of science, engineering and technology

Appendix D: Informed consent form

UNISA college of science, engineering and technology

Informed consent form

My name is Mr. Tawanda Blessing Chiyangwa, a PhD student in the School of Computing at the University of South Africa. I am conducting a survey to evaluate the main success factors of agile software development projects in South Africa through a questionnaire, working under the supervision of Prof Ernest Mnkandla. I am kindly seeking your permission to include you (IT, ICT, software developer, IT managers and ICT managers regarded as agile professionals) as participants in this research. Please read this consent document carefully. If you grant permission to participate in this study, please sign the agreement at the end of the form and return it to me.

Title of the research project:

An empirical study of success factors in agile software development projects in South Africa

Please take note of the following:

1. IT, ICT, software developer, IT managers and ICT managers are required to answer the questionnaire.
2. Data and information I share will be handled confidentially and anonymously.

- Data, information and references will be protected as required by the Data Protection Act of South Africa.
- Your name will not be associated with any data that are collected during this survey.

Time required:

We expect a session to last about 30 minutes. Participants will answer structured and semi-structured questions. Each participant is required to answer all the questions. The data collected will be used for research purposes only.

University of South Africa
Preller Street, Muckleneuk Ridge, City of Tshwane
PO Box 392 UNISA 0003 South Africa
Telephone + 27 12 429 6933 Facsimile + 27 12 429 6848

UNISA college of science, engineering and technology

Risks:

There are no known risks associated with this study.

Finally, we greatly appreciate your time and effort in participating in this survey. Remember, the research does not affect your work. Please do not hesitate to ask if you have any questions regarding the experiment.

The rights participants are as follows:

- You are voluntarily taking part in this study and it is your right not to participate.
- You can withdraw from this study at any time and have the information provided in your questionnaire removed in its entirety from this study.

My contact details are as follows:

Email address: chiyangwa.tawanda@gmail.co.za

Cell number: +27743663721

Agreement:

Your signature below indicates that you have read this consent form in its entirety and that you are voluntarily participating.

Venue:

The survey will take place online in South African organisations.

Surname: _____________________ Cell Number: __________________

Name: _______________________ Date: _________________________

Signature: _____________________

University of South Africa
Preller Street, Muckleneuk Ridge, City of Tshwane
PO Box 392 UNISA 0003 South Africa

Appendix E:Themes of the success factors from interview questionnaire - appendix B

Category	Success Factors	Frequency of Appearance in Interview
Process Factors	Proper Management Process	14
	Regular Working Schedule	15
	Good Communication	13
	Customer involvement or Participation	10
Organisational Factors	Proper Agile-Style Environment	14
	Project Team is Collocated	15
	Organisational Cooperative	12
	Organisational Culture	9
	Sponsor Commitment	8
	Agile Methodologies is Universally Accepted	5
Performance Expectancy Factors	Agile Software Development Projects is Useful	6
	Agile Software Development Projects enables to Accomplish Tasks	7
	Agile Software Development Projects increases my Productivity	8
	Increase Chances of Promotion	9
Actual Success Factors	Cost Being Under Budget	12
	Timeliness of Project Completion	11
	Scope of Project Being Met	14
	Quality of Project Outcome	13
	Terms of Requirements of Project Met	12
	Plans Intended Accomplished	5
	Plans Intended for the Project to be Accomplish	9
Project Factors	Agile-Friendly Project Type	5
	Appropriate Project Size	8
	Proper Project Cost Evaluation and Risk Analysis	9
People Factors	Individual Motivation and Expertise	9
	Educated, Light-touch and Adaptive	11
	Coherent and Self-organising Teamwork	12
	Good Customer Bond	14
Technological Factors	Proper Agile Software Engineering Practice	14
	Appropriate Technical Training	13

Category	Success Factors	Frequency of Appearance in Interview
	Correct Integration	12
	Delivery Strategy	14
Intentional Behaviour Factors	Plan Practice Agile Software Development Projects	14
	Effort Practice Agile Software Development Projects	13
	Intend Practice Agile Software Development Projects	12
	Predict Practice Agile Software Development Projects	11
	Intend Use Agile Method	10
	Predict Use Agile Method	9
	Intend Use Agile Methods on a Regular Basis	11
Effort Expectancy Factors	Understandable	12
	Easy to Use	13
	Learning is Easy to Use	14
	Easy to Use Skilful Agile Techniques	10
	Easy to Use Agile than Traditional Methodologies	12
	Practicing to Use Agile Method is Easy to Use	11
	Clear	13