UNIVERSITY OF KWAZULU-NATAL

Critical success factors that influence the performance of agile software development methodologies in organisations

By

Yeshmeeta Peters

983170355

A dissertation submitted in fulfillment of the requirements for the degree of Master of Commerce Research in Information Systems and Technology

College of Law and Management Studies

School of Management, Information Technology and Governance

Supervisor: Mr. Mudaray Marimuthu

2020

DECLARATION

I, **Yeshmeeta Deodutt Peters** declare that

i (i) The research reported in this dissertation/thesis; except where otherwise indicated, is my original research.

ii (ii) This dissertation/thesis has not been submitted for any degree or examination at any other university.

iii (iii) This dissertation/thesis does not contain another person's data, pictures, graphs, or other information unless specifically acknowledged as being sourced from other persons.

iv (iv) This dissertation/thesis does not contain other persons' writing unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:

a) their words have been re-written, but the general information attributed to them has been referenced.

b) where their exact words have been used, their writing has been placed inside quotation marks and referenced.

(v) Where I have reproduced a publication of which I am author, co-author, or editor, I have indicated in detail which part of the publication was actually written by myself alone and have fully referenced such publication.

(vi) This dissertation/thesis does not contain text, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the dissertation/thesis and the References sections.

Signature:

Date: 15 January

2021

DEDICATION

I dedicate this dissertation to God who has guided me throughout my life and filled my heart with love, knowledge, gratitude, and humility.

To my late father, Prem Gopi who has been my strength, my inspiration, and who is with me every step of my life. I strive to make you proud and I know you are looking down upon me and showering me with the love and strength that I needed when I thought I could not manage life without you.

To my mum (ma) who is still with me and showering me with the love and care that only a mother knows. She has always spurred me on to do my best and be the best version of myself.

To my dear husband and best friend for his ongoing love, care, and support during a trying time in the world. I appreciate all that you do.

To my heartbeats, my two babies, Rahul & Anjali who have no idea what I have been doing but are a constant source of love and inspiration to me – I love you both immeasurably.

ACKNOWLEDGEMENT

A special thank you to my husband, children, and mum for supporting me and understanding me during this time.

Thank you to my husband, Amille Peters, my babies Rahul and Anjali who wondered what I was always busy with during odd hours of the night and often found me awake through the night. Thank you all for your understanding, patience, and for allowing me to complete this study.

I would like to extend a special thank you to my supervisor, Mudaray Marimuthu (Ashley) for understanding my unearthly hours of working and guiding me through the research. I appreciate your time and assistance.

Thank you for your support during this scary time in the world…

God bless you all with health, happiness, safety, and prosperity.

ACRONYMS

CSF	Critical Success Factor
DSDM	Dynamic Systems Development Method
RAD	Rapid Application Development
SAFe	Scaled Agile Framework
SDLC	Software Development Life Cycle
SPSS	Statistical Package Social Sciences
TSDMs	Traditional Software Development Methods
XP	Extreme Programming
AUP	Agile Unified Process
UP	Unified Process
RUP	Rational Unified Process
USDP	Unified Software Development Process

ABSTRACT

The agile manifesto was brought into existence in 2001 and agile as a methodology was derived in the 1990s. The reason for the formulation of this methodology was to create methods to produce software in a better manner that could fulfill the customer's needs in an environment that was iterative and controlled. The types of agile methodologies being followed are Scrum, extreme project management, adaptive project management, and dynamic project management method and scrum is the most widely utilized. There is insufficient research into the hierarchy of importance of the critical success factors that affect agile projects. Critical success factors of organisational structure, people, process, technical and, project factors have been identified in previous studies, however, the ranking of these factors in terms of the level of importance for agile success has not been studied enough. These critical factors are classified as Technical, Organisational, Process, Project, and People categories. There were suggestions from researchers that test automation and cloud computing can also positively affect the success of a project using agile. Since these two factors were not studied in conjunction with the other critical factors mentioned previously, this study extended previous studies by incorporating these factors. This study expanded the factors by including cloud computing and test automation as possible critical factors to the successful implementation of agile software development. The research method chosen for this study was the quantitative method. The data was collected using questionnaires and was analyzed via descriptive and inferential statistics. To achieve an acceptable statistical power, a sample size of 200 agile practitioners was targeted, but the researcher was able to obtain 110 responses. SPSS version 27.0 was used for the descriptive and inferential data analysis and the statistical tests. The main findings indicated that people, technical factors, and test automation were the top three critical success factors in terms of importance. The project, people, and organisational structure were the top three critical success factors in terms of performance. Cloud computing was found to be less important whereas test automation was found to be an important factor for agile success. Significant gaps were identified between the critical success factors and their performance in organisations. The study recommends that organisations place additional emphasis on the critical success factors that affect agile success and the performance of these factors to close the gap identified in this study. Further recommendations are to provide adequate training in agile processes.

TABLE OF CONTENTS

DECLARATION ii
DEDICATION iii
ACKNOWLEDGEMENT iv
ACRONYMS v
ABSTRACT vi
LIST OF TABLES xi
LIST OF FIGURES xii
CHAPTER 1 1
INTRODUCTION TO THE STUDY 1
1.1 INTRODUCTION 1
1.2 BACKGROUND TO THE STUDY 1
1.3 RESEARCH PROBLEM 3
1.4 AIM OF THE STUDY 5
1.5 RESEARCH OBJECTIVES 5
1.5.1 Research Questions 5
1.5.2 Objectives 5
1.6 SIGNIFICANCE OF THE STUDY 6
1.7 JUSTIFICATION OF THE STUDY 7
1.8 OVERVIEW OF THE STUDY 7
1.9 CONCLUSION 8
CHAPTER 2 9
LITERATURE REVIEW 9
2.1 INTRODUCTION: SOFTWARE DEVELOPMENT IN GENERAL 9
2.2 TRADITIONAL METHODOLOGIES 9
2.2.1 Code and fix model 11
2.2.2 Waterfall Model 12
2.2.3 Prototype Model 14
2.2.4 Incremental model 16
2.3 AGILE METHODOLOGIES 18
2.3.1 Extreme Programming 20
2.3.2 Scrum 21
2.3.3 Scaled Agile Framework (SAFe) 23
2.4 AGILE METHODOLOGIES VERSUS TRADITIONAL METHODOLOGIES 25
2.5 CRITICAL SUCCESS FACTORS 28

2.5.1 Organisational Factors ... 30
2.5.2 People Factors ... 31
2.5.3 Process Factors ... 32
2.5.4 Project ... 33
2.5.5 Technical Factors ... 34
2.5.6 Test Automation ... 34
2.5.7 Cloud ... 35
2.6 The Study's Conceptual Model ... 36
2.7 CONCLUSION ... 37
CHAPTER 3 ... 38
RESEARCH METHODOLOGY ... 38
3.1 INTRODUCTION ... 38
3.2 RESEARCH DESIGN ... 38
3.3 RESEARCH PHILOSOPHY/APPROACHES ... 39
3.4 SURVEY DESIGN AND LAYOUT ... 40
3.5 STUDY SITE AND TARGET POPULATION ... 41
3.6 APPROACH TO THEORY DEVELOPMENT ... 42
3.7 METHODOLOGICAL CHOICE: (RESEARCH APPROACHES) ... 43
3.8 DATA COLLECTION STRATEGY ... 45
3.8.1 Sampling Strategies ... 45
3.8.2 Sample Size and Sample ... 46
3.9 DATA COLLECTION METHODS ... 47
3.10 DATA QUALITY CONTROL ... 48
3.11 ETHICAL CONSIDERATIONS ... 50
3.12 DATA ANALYSIS ... 51
3.13 CONCLUSION ... 53
CHAPTER 4 ... 55
RESULTS ... 55
4.1 INTRODUCTION ... 55
4.2 DESCRIPTIVE STATISTICS ... 55
4.2.1 Demographic factors of respondents ... 55
4.2.2 Type of agile environment ... 56
4.2.3 Organisation Location ... 58
4.2.4 Number of members in the agile project team ... 58
4.2.5 Number of agile projects (agile maturity) ... 58
4.2.6 Organisation ... 58

4.2.7 Respondent Role & experience 60
4.2.8 Agile project experience 60
4.3 CRITICAL SUCCESS FACTORS 61
4.3.1 Descriptive analysis of importance factors 61
4.3.2 Critical success factors that are perceived to be most important by stakeholders for agile success. 70
4.4 OVERALL RANKING OF CSFS 75
4.5 DESCRIPTIVE ANALYSIS OF THE PERFORMANCE OF SUCCESS FACTORS IN AGILE PROJECTS 76
4.5.1 Organisational Structure 76
4.5.2 People 78
4.5.3 Process 79
4.5.4 Technical Factors 81
4.5.5 Project 82
4.5.6 Test Automation 83
4.5.7 Cloud Computing 84
4.6 RII AND RANKING OF PERFORMANCE FACTORS 85
4.6.1 Organisational Structure 85
4.6.2 People 86
4.6.3 Process 87
4.6.4 Technical Factors 88
4.6.5 Project 88
4.6.6 Test Automation 89
4.6.7 Cloud Computing 90
4.7 OVERALL RANKING OF PERFORMANCE CSFS 91
4.8 THE GAP BETWEEN THE PERCEIVED IMPORTANCE AND ACTUAL PERFORMANCE OF SUCCESS FACTORS 91
4.8.1 Reliability Analysis 91
4.8.2 Paired Sample T-Test: 93
4.9 CONCLUSION 96
CHAPTER 5 .... **97**
DISCUSSION, RECOMMENDATIONS, AND CONCLUSION **97**
5.1 INTRODUCTION 97
5.2 DISCUSSION 97
5.3 RECOMMENDATIONS 101
5.4 LIMITATION OF THE STUDY AND FUTURE RESEARCH 103
5.5 CONCLUSION 104

REFERENCES...106
APPENDICES...116
APPENDIX A: INFORMED CONSENT 117
APPENDIX B: CONSENT TO PARTICIPATE 119
APPENDIX C: RESEARCH INSTRUMENT (QUESTIONNAIRE) 120
APPENDIX D: ETHICAL APPROVAL LETTER 131

LIST OF TABLES

Table 4.1 Demographical statistics of respondents....55
Table 4.2 Demographic statistics of Agile development, Location, Number of project team members, Number of projects....56
Table 4.3 Demographical statistics of Organisation, Role, and years of experience....58
Table 4.4 RII and Ranking of Organisational Structure CSFs....70
Table 4.5 RII and Ranking of People CSFs....71
Table 4.6 RII and Ranking of Process CSFs....71
Table 4.7 RII and Ranking of Technical Factor CSFs....72
Table 4.8 RII and Ranking of Project CSFs....73
Table 4.9 RII and Ranking of Test Automation CSFs....74
Table 4.10 RII and Ranking of Cloud Computing CSFs....75
Table 4.11 RII showing CSFs ranked by Mean RII on Perceived Importance....75
Table 4.12 RII and Ranking of Organisation Structure CSFs....85
Table 4.13 Performance: RII and Ranking of People CSFs....86
Table 4.14 Performance: RII and Ranking of Process Factor....87
Table 4.15 Performance: RII and Ranking of Technical Factor CSFs....88
Table 4.16 Performance: RII and Ranking of Project CSFs....89
Table 4.17 Performance: RII and Ranking of Test Automation CSFs....89
Table 4.18 Performance: RII and Ranking of Cloud Computing CSFs....90
Table 4.19 RII showing CSFs ranked by Mean RII on Perceived Performance....91
Table 4.20 Cronbach - Reliability Analysis - Importance....92
Table 4.21 Cronbach - Reliability Analysis - Performance....92
Table 4.22 Paired Samples T-Test....93
Table 4.23 Summary of Hypothesis Testing for CSFs....95

LIST OF FIGURES

Figure 1.1 Project Failure Rates by Project Size 6
Figure 2.1 Software development life cycle 10
Figure 2.2 Waterfall model 14
Figure 2.3 Prototype model 14
Figure 2.4 Incremental model diagram showing versions 16
Figure 2.5 The Scrum life cycle 21
Figure 2.6 The XP Release Cycle .... **Error! Bookmark not defined.**
Figure 2.7 Scaled Agile Framework 25
Figure 2.8 Failure and success of agile and waterfall projects from 2013-2017 28
Figure 2.9 Conceptual Framework for the study 37
Figure 3.1 The Research Onion 39
Figure 3.2 The Deductive Approach 43
Figure 4.1 Descriptive Statistics for Organisational Structure 61
Figure 4.2 Descriptive Statistics for People 63
Figure 4.3 Descriptive Statistics for Process 64
Figure 4.4 Descriptive Statistics for Technical Factors 65
Figure 4.5 Descriptive Statistics for Project 67
Figure 4.6 Descriptive Statistics for Test Automation 68
Figure 4.7 Descriptive Statistics for Cloud Computing 69
Figure 4.8 Descriptive Statistics for Organisational Structure 77
Figure 4.9 Descriptive Statistics for People 78
Figure 4.10 Descriptive Statistics for Process 80
Figure 4.11 Descriptive Statistics for Technical Factors 81
Figure 4.12 Descriptive Statistics for Project 82
Figure 4.13 Descriptive Statistics for Test Automation 83
Figure 4.14 Descriptive Statistics for Cloud Computing 84
Figure 5.1 Research framework of critical success factors 100

CHAPTER 1
INTRODUCTION TO THE STUDY

1.1 INTRODUCTION

Software is a pivotal part of society today, whether it is the use of cellphones, tablets, laptops, or other devices. It has a positive impact on business by reducing the time spent on manual tasks and offering a return on investment by being a service to individuals and organisations. Organisations invest money in software and if the software fails, the financial implication is huge. Software applications used in social media have become an important mechanism for communication between organisations, schools, families, and other users. The software industry has been growing due to the increased demand for software products (Mohammed & Abushama, 2013).

The creation of software entails adhering to an application development process (Harisha, 2018). It is necessary to choose and follow a structured process to develop software to create a software solution and to prevent time wastage, money wastage, and unhappy developers (Harisha, 2018). Agile methodologies are gaining popularity and are being utilized more by software and information systems practitioners (Mohammed & Abushama, 2013). Agile methods are also popular by reducing development expenses and managing changes that are required at later phases in the project (Jalali & Wohlin, 2010). Mbelli & Hira (2016) showed how agile methodologies were being viewed in South Africa with an acceptance rate of 50%. The survey of this paper sought out to gather information on how agile methodologies were perceived by these professionals, their roles, their experience using agile in developing software, their education, and their understanding of why agile methods fail or succeed (Mbelli & Hira, 2016). It was noted from the findings that the ratio of the positive versus the negative encounters with agile processes was equal.

1.2 BACKGROUND TO THE STUDY

Software development dates back to the 1950s and that was when the software development process was recognized to be the only formal method being used (Oleksandrova, 2018).

Around 1960, the main aim of this methodology was to create large business applications in an era of large business organisations (Oleksandrova, 2018). In 2001, the Agile Manifesto was developed by 17 people who wanted to recognize other approaches to software development than the heavyweight methodologies that were considered to be rigid and focused on documentation (Awad, 2005). According to Bhardwaj (2014), the objective of the agile methodology was to focus on people, their relationships, and functional deliverables instead of intensive paperwork. Further, it aimed to focus on the working relationship between the client and the project team instead of deciding on service level agreements and being able to adapt to amendments instead of focusing on adhering to a set agenda (Bhardwaj, 2014).

Agile, which is a software development methodology has its main concern with the satisfaction of the customer by continuously delivering software increments (Bhardwaj, 2014). In this way, the customer is kept abreast of changes and is constantly involved in the project. Hence, a characteristic of an agile project is that it creates a deliverable solution from a minimum number of requirements, and it is focused on minimizing the scope of the project (Bhardwaj, 2014). Agile project development focuses on quick, rapid development with the main emphasis on incremental development (Nerur, Mahapatra, & Mangalara, 2005). By minimizing the scope of the project and keeping the customer satisfied at each iteration, agile projects can deliver successful outputs in a shorter timeframe after each iteration. As noted by Bhardwaj (2014) and Nerur et al. (2005), agile projects can ensure customer involvement and can deliver increments quickly due to the smaller scope and smaller iterations. These shorter increments are more likely to be successful as the scope is smaller and can be changed quickly to meet the customer's requirements.

Agile methods succeed by the dedication of people, their teamwork, ability to adapt to changing circumstances, and the concept of sharing. Organisations must be prepared and have the necessary tools, techniques, culture, people, and resources to adopt agile methods (Nerur et al., 2005). Organisational culture influences the decision, strategies, and approaches, and organisations that have been successful in the traditional approach of software development for many years would be reluctant to adopt a new approach if it is not entirely necessary (Nerur et al., 2005). The barriers to the adoption of agile include not being able to alter an organisation's culture which is followed by resistance to change and then attempt to place agile into a non-agile framework (Bhardwaj, 2014). Further, cultural barriers and teams that are globally distributed also contribute to the resistance to agile adoption. According to the 2018 Standish Group CHAOS Report, agile projects are two times more likely to be successful than traditional projects, but agile projects still fail (Mersino, 2018). Agile projects fail due to inexperience with agile methodologies, insufficient understanding of the change required to implement agile, and the organisational culture imbalance with agile (Miller, 2013).

This study will determine success factors by measuring software developers' perception of important factors based on their work experience in an agile environment and then measure their perception of the performance of these factors in the agile environment. The study aims to rank the critical success factors that influence the outcome of agile projects in order of importance. This will help determine the gaps that exist in software development companies concerning the implementation of success factors in organisations.

1.3 RESEARCH PROBLEM

Project success is defined as a solution that is delivered and adheres to the success criteria within a reasonable range and failure is viewed as the project being unable to deliver a solution (Miller, 2013). Some elements contribute to the positive outcome or downfall of projects utilizing agile methods. Critical success factors recognize the most necessary aspects that require assistance for an organisation to be successful (Yaghoobi, 2018). Work on the factors positively influencing the use of agile methodologies are covered by numerous studies (e.g. (Hameed, Latif, & Kholief, 2016); (Chiyangwa & Mnkandla, 2017);(Chow & Cao, 2008); (Misra, Kumar, & Kumar, 2009); (Stankovic, Nikolic, Djordjevic, & Cao, 2013)). Further, Aldahmash, Gravell & Howard (2017) reviewed the critical success factors that were identified in research from 2006 to 2016. The eight factors identified were, the plan of delivering the software, the team's abilities and learning, the agile procedures, client participation, managing the project, the culture of the organisation, interaction, and the guidance of higher management (Aldahmash et al., 2017).

The chosen critical factors have been divided into a range of factors which incorporate Technical, Organisational, Process, Project, and People categories. Aldahmash et al. (2017) indicated that future work can be conducted to investigate the relevance of these critical success factors and the weighting of every factor. Further, the majority of the previous research determine success factors by measuring project success and then using regression analysis to determine which factors influenced this success. In addition to the factors highlighted above, there are suggestions from researchers that test automation (Collins & de Lucena, 2012) and cloud computing (Younas, Jawawi, Ghani, Fries & Kazmi, 2018) can also positively affect the success of a project using agile. Since these two factors were not studied in conjunction with the other success factors mentioned previously, this study extended previous studies by incorporating these factors.

The paper by Collins and de Lucena (2012) presented the results of implementing Test Automation in projects that used Scrum as the development process and open source testing tools. Collins and de Lucena (2012) observed that implementing test automation could be very beneficial when the team has the motivation to adopt the correct ways of working. The following lessons were noted: (Collins & de Lucena, 2012). According to Collins and de Lucena (2012), test automation tools should be simple enough for all members to use. Using test automation for documentation and for observing test execution history can be valuable to the project and the test automation tools should be simple for all project members to use (Collins & de Lucena, 2012). It was evident from the paper that team collaboration and test automation are important in the success of agile projects. The above findings highlighted the need to investigate test automation as an additional critical success factor in agile projects.

The systematic literature review by Younas et al. (2018) indicated that cloud computing does not change the agile software development process, however, cloud computing positively influences agile development. One positive influence is that cloud computing resolves communication issues among team members, while transparency is increased by the clear implementation of the user story, management of the backlog, and traceability (Younas et al., 2018). Another positive aspect is that cloud computing offers a reduction in the market delivery of software and cloud computing services reduce the cost incurred for server management by hardware engineers (Younas et al., 2018). The above influence of cloud computing on agile projects emphasized the need to include cloud computing as an additional critical success factor in this study of agile projects.

Further, there is insufficient research into the hierarchy of importance of the critical success factors that affect agile projects. Critical success factors have been identified in previous studies, however, the ranking of these factors in terms of the level of importance for agile success has not been studied enough. This study used previously identified success factors which was expanded by adding test automation and cloud computing factors and investigating the gap between the importance and performance of these critical success factors.

1.4 AIM OF THE STUDY

This research study aims to establish a hierarchy of critical success factors that affect agile success and the related performance of these factors in agile projects.

1.5 RESEARCH OBJECTIVES

1.5.1 Research Questions

Based on the problem statement above the research questions are:

Research Q1: Which critical factors are perceived to be most important by stakeholders for agile success?

Research Q2: What is the performance of these CSFs in agile projects?

Research Q3: What is the gap between the perceived importance and the actual performance of the CSFs?

1.5.2 Objectives

Based on the research questions above the following objectives were established:

Objective 1:

To identify which critical factors are perceived to be most important by stakeholders for agile success.

Objective 2:

To determine the performance of critical success factors within agile teams.

Objective 3:

To use the findings from objectives 1 and 2 above to understand the link between the perceived critical factors and their actual performance.

1.6 SIGNIFICANCE OF THE STUDY

This research will contribute to the knowledge and research available on agile methodologies and specifically the success of agile projects. The findings of this study will enable agile practitioners and organisations to focus on the critical success factors with a particular emphasis on the hierarchy of importance.

This will allow critical success factors to be at the forefront of agile projects. The contribution of this study will be in the form of awareness and empowerment as agile practitioners should be able to ensure that the critical success factors perform in alignment with their importance. It will assist in ensuring greater success in agile projects. At the start of an agile project, agile practitioners should be formulating a strategy that aligns the critical success factors with their performance.

Figure 1.1: Project Success Rates by Project Size. Source: Standish Group (https://vitalitychicago.com/blog/agile-projects-are-more-successful-traditional-projects/)

Figure 1.1 shows the Standish Group Chaos Study results which indicate that although the probability of agile projects to be successful is twice as much as waterfall projects, they do fail as well (Mersino, 2018). The Standish Group found that failure rates are higher in larger projects. Agile projects do fare better than waterfall projects for different sizes of projects. However, the chart shows that the project size has a greater influence on project failure even when using the agile approach (Mersino, 2018). To decrease agile project failure, project teams should focus on the critical factors that contribute to their success.

1.7 JUSTIFICATION OF THE STUDY

This study aims to identify the hierarchy of the critical success factors that affect projects utilizing the agile approach, concerning their priority. Although many factors positively influence agile success, certain factors are pivotal to their success. The weighting of the critical success factors will assist practitioners in focusing their attention on the factors in order of priority. Further, understanding the performance of these factors will assist organisations in understanding the areas that need attention to ensure agile project success. With the advancement of technologies, cloud computing and test automation are important to agile development success but were not studied together with other factors documented in the literature. This study will expand the factors by including cloud computing and test automation as possible success factors to agile software development.

1.8 OVERVIEW OF THE STUDY

This thesis comprises five chapters which are as follows:

Chapter one:

Chapter one is the introductory chapter of the research study and provides the background of the research. It also presents the research questions and objectives. It provides a summary of the important points of the thesis and the research outline.

Chapter two:

This chapter is the Literature Review which forms the basis of the literature that the researcher presents on the topic. It discusses traditional methodologies, agile methodologies, and a comparison is shown. It sets the tone for agile project success and the critical success factors that are integral to the research topic.

Chapter three:

This chapter is the Research Methodology and presents the research methods that are available and details the methods used by the researcher and the reasoning behind the methodology choice.

Chapter four:

This chapter is the Results and presents the results obtained via the research tool which was a questionnaire. The relevant tables from the statistical tool used which was SPSS 27 are presented and explained.

Chapter 5:

This chapter includes the discussion, recommendations, and conclusion. This chapter is the closing chapter of the entire research and discusses the findings of the research study and the conclusions are drawn. It also discusses the contribution of this study to the field of information systems and agile development. The limitations are discussed and how the researcher attempted to deal with them.

1.9 CONCLUSION

This chapter is the first chapter of this paper and introduces the research study and provides a breakdown of the subsequent chapters. The background, the research problem, research objectives, research questions, and the research tool were presented. The significance and justification of the study were also outlined. This study has included two additional success factors which are test automation and cloud computing which were indicated in this chapter. This will contribute to the knowledge and research available on agile projects. Chapter 2 examines the literature on agile methodologies.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION: SOFTWARE DEVELOPMENT IN GENERAL

Software development is an essential part of modern industry. People are more reliant on technology and hence the reliance on software to be developed to meet the requirements of consumers and users across the world. Computers and software are a vital part of industry and technology and organisations have become dependent on software for office work, administration, banking, and other sectors (Munassar & Govardhan, 2010). The goal of software engineering is to develop work that creates software programs that are of exceptional quality (Munassar & Govardhan, 2010). The software development model is the basis of the entire software development cycle that entails the core structure that feeds into the tasks that need to take place for the software development to be implemented (Yu, 2018). Bhatnagar (2015) describes software project development as one of the most pivotal areas in the world of software. The software development life cycle (SDLC) provides the framework of all the tasks that are required to be implemented during the software development process (Bhatnagar, 2015). SDLC focuses on the quality of the software by minimizing the failure and potential risk in the project (Bhatnagar, 2015). Waterfall SDLC is categorized as the traditional SDLC and agile SDLC models as the modern SDLC which will be discussed (Leau, Loo, Tham & Tan, 2012).

2.2 TRADITIONAL METHODOLOGIES

Boehm (2002) described traditional software development methods (TSDMs) as being thoroughly planned, codified processes, extensive reuse, detailed documentation, and in-depth design at the start of the project. A constantly changing world has a requirement that produces adaptive software applications, and the rigid traditional methodologies are unable to support the dynamic nature of project requirements (Arikpo & Osofisan, 2019). Popli, Anita, and Chauhan (2013) stated that traditional methodologies are schedule-driven and follow the schedule throughout the duration of the project. The model follows a step by step sequence which starts with the feasibility study, gathering requirements, producing the solution, testing, and deployment (Ahmad, Soomro & Naqvi, 2016)

According to Popli et al. (2013), the traditional approach confirms that complex software can be developed end to end without going back to the requirements, but the common concern is that there is a possibility that requirements do change, and the traditional approach does not account for it. This methodology is also referred to as heavyweight as it entails a heavy structure and extensive work upfront (Javanmard & Alian, 2015). This method requires a comprehensive design strategy, and documentation is necessary at the outset (Ahmad et al., 2016). Some software practitioners viewed this methodology as time-consuming and frustrating as it did not allow for changes (Javanmard & Alian, 2015). The following are some common traditional software methodologies.

The Software Development Life Cycle (Figure 2.1) is a process that explains the events that occur in each phase of the software development process. It shows how software is developed through the phases, the maintenance, and the replacement of software if necessary (Stoica, Mircea, & Ghilic-Micu, 2013)

Figure 2.1: Software development life cycle (Stoica et al., 2013)

2.2.1 Code and fix model

This model which is also known as Cowboy coding is a cyclic process that comprises two steps which are Coding and Fixing (Mkrtchyan, 2020). There is no well-defined planning and the vital role players which are known as cowboy coders are the software engineers who are responsible for the work with minimal or no collaboration with the business, design, or other teams (Mkrtchyan, 2020). The coding process starts after some communication with the product team and fixing occurs as the project progresses so that the team can fix bugs during the course of the project (Mkrtchyan, 2020). The methodology is not properly defined and is a basic two-step process (Grubb & Takanga, 2003). The first step is to develop the code and the next step is to fix or change the code. There is no allowance in this method to analyze, design, or follow the software development life cycle or any structured process (Grubb & Takanga, 2003). This model is still used in the software development industry as when a change is required very quickly, there is insufficient time for analysis, design, feasibility studies, or proper testing (Grubb & Takanga, 2003). The order of this model is to code and to reflect on the requirements, design, testing, and maintenance at a later stage (Boehm, 1988).

Advantages of this model:

1. The code and fix model has little to no overhead with regards to documentation, design preparation, quality checks, and other coding processes (Boehm, 1988)

2. The code and fix model requires no training as the software developer can start to code immediately without having to be trained in processes or specific requirements (Boehm, 1988). This will save the organisation money and time resources as training costs can be reduced.

3. The model can be used in cases where changes are required quickly and there is no time for extensive documentation, analysis, design, or software development life cycle processes (Grubb & Takanga, 2003). This will also save time as documentation is not required.

Disadvantages of this model:

1. The subsequent fixes became expensive as the code became poorly structured over time (Boehm, 1988). This occurs as the fixes were implemented quickly and lose their structure due to time constraints.

2. The match between properly designed software and the requirements of the users was not good as the requirements were not properly defined (Boehm, 1988). The requirements have to address the needs of the users and require being documented appropriately.

3. The code became costly to fix as there was insufficient preparation for testing and possible changes. This showed that these stages needed to be clearly defined and the planning and preparation were necessary early on (Boehm, 1988). There was not enough time for testing making changes.

4. The main issue with this model is that it does not allow for changes and is rigid (Grubb & Takanga, 2003). Being rigid does not allow for changing needs.

5. There is no acknowledgment in the model to provide alternatives that show that one route may be less risky or cost less than the other (Grubb & Takanga, 2003).

6. The lack of properly defined stages does not allow for the anticipation of problems (Grubb & Takanga, 2003). This shows that the model cannot recognize issues quickly

7. The model does not allow for valuable documentation that could assist in future development (Boehm, 1988). The lack of documentation will prevent other developers to have a document to reference as a guide of functionality or rules.

Due to the disadvantages highlighted, this model cannot be used in large-scale software development in which the requirements, design, and risk identification are evaluated first (Boehm, 1988).

2.2.2 Waterfall Model

Thummadi, Shiv, Berente, & Lyytinen (2011) describe the waterfall model as a one-directional, top-down, and non-iterative process for designing and developing software. The waterfall model which is known as the classical model entails planning that occurs early in the project before development begins (Munassar & Govardhan, 2010). The stages do not overlap and start with defining the requirements and then go to the design, coding, testing, and end with maintenance which allows this model to form the basis for other models (Munassar & Govardhan, 2010). The requirements must be known at the outset and the model comprises of different phases of development which the requirements analysis, designing, coding, testing, and maintenance, while each phase has to be completed before the next phase begins (Yu, 2018).

It is difficult to anticipate all details before a project starts as additional details are discovered when the project continues (Yu, 2018). The main disadvantage of this model is evident by issues in the software that are not identified until late in the testing phase which does not allow enough time for correction and could result in major issues that hinder the project schedule and increase the cost (Yu, 2018).

The Waterfall methodology indicates a sequential process between stages that are defined at the outset (Javanmard & Alian, 2015). The first stage which is the requirements analysis entails establishing what is required for the project and how long it will take to achieve (Leau et al., 2012). The next stage is the design and architectural planning where diagrams are produced to show technical infrastructure. The project then proceeds to the development stage in which coding is done until the build fulfills the requirements (Leau et al., 2012).

The third stage is the development stage where coding could be distributed among different teams according to skills required. Testing is the fourth stage, and it overlaps with the development stage to address any issues that may be exposed. The fifth stage is the release and occurs once the customer is satisfied (Leau et al., 2012).

Advantages of the waterfall model are: (adapted from Rather & Bhatnagar, 2015)

1. It is used in simple projects with fixed deadlines
2. The requirements are defined before the design.
3. It is easily understood.

Disadvantages of the waterfall model are: (adapted from Rather & Bhatnagar, 2015)

1. The issues are only discovered during testing.
2. It represents an ideal project which is sometimes unrealistic.
3. It is difficult to implement any changes.
4. The requirements are not very clear and the delivery could be late.

Each stage produces related documents and deliverables that are used to start the next stage (Ahmad et al., 2016). The term waterfall is generally used as a name for all sequential development in engineering (Awad, 2005). Figure 2.2 shows the Waterfall Model which shows the waterfall-like flow.

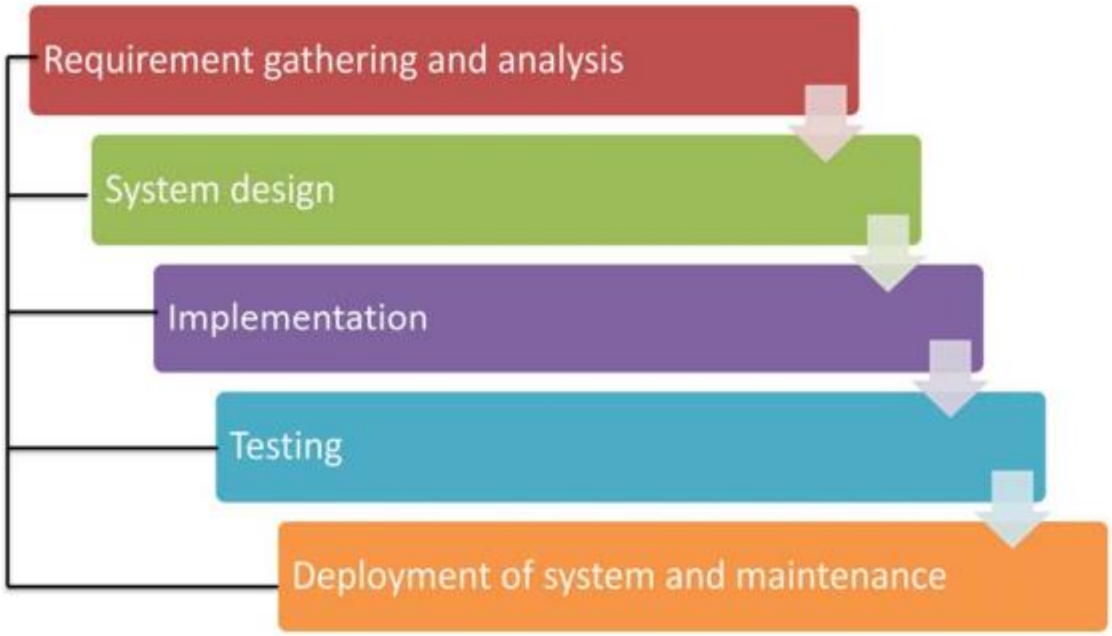

Figure 2.2 Waterfall Diagram (Stoica et al., 2013)

2.2.3 Prototype Model

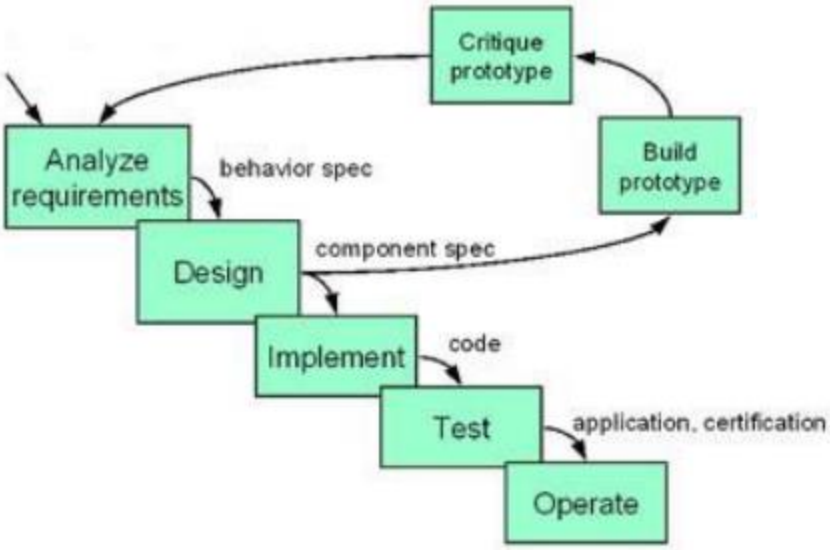

Figure 2.3 Prototype model (Khan & Jamal, 2017)

Prototyping is a mechanism for extracting, presenting, and refining the needs of the user by creating a working model of the final system which is done quickly and properly (Bernard Boar, 1993). Figure 2.3 shows that the prototype model starts with the analysis of the requirements, follows with the quick development of the prototype, user testing and feedback which is done by the customer, and the refinement or revision of the prototype is an iterative process before the model ends with the release of the final product. Bernard Boar (1993) introduced this model as being able to uncover solutions as the models are incrementally refined which can eventually solve the definition problem. In the software development life cycle, the implementation occurs after the design and specification, while in the prototype model, the implementation of a prototype occurs as quickly as the functional model is working (Pomberger & Weinreich, 1994).

Rather and Bhatnagar (2015) described the prototype model as an iterative type of model by which the developer can create progressively more complete software versions and this model is therefore classed as an evolutionary process model. This model has a quick design phase in which the developer designs the software after the requirements are gathered (Rather & Bhatnagar, 2015). It is this quick development and testing of functional models that are known as prototypes of new systems (Susanto & Meiryani, 2017). The prototype will be changed constantly according to the feedback from users until they are satisfied (Yu, 2018). The developers must be experienced so that they can identify the most important requirements when creating the prototype (Yu, 2016). Further, the involvement and input of the user are important, and the prototypes are built to provide proof of the functioning (Javanmard & Alian, 2015).

When the scope of the project is outlined in the Requirements Planning stage, the users communicate with the analysts to create prototypes in the User Design stage (Awad, 2005). Feedback is given in the Construction stage by the users, while in the Cutover stage the users are equipped to utilize the system as it is implemented (Awad, 2005).

Some advantages of the prototype model are: (adapted from Awad, 2005)

1. It allows for early visibility. This results from the quick delivery of the prototypes.
2. It also allows for early design.
3. Customers can use and get a feel for the system and provide feedback.
4. It reduces costs. This is possible as the prototype can be changed before the final system implementation.

Some disadvantages of the prototype model are: (adapted from Awad, 2005)

1. Management becomes complicated due to various prototypes being implemented
2. Possibility of systems that are not completed. This results from the possibility that prototypes will be delivered, but not resulting in a final product.
3. There is a possibility of a system that does not meet all the requirements. This stems from a working solution being developed, but not the best possible solution.
4. It is not suitable for large systems. This is due to the complicated process that will result if the system is large as prototypes would be difficult to implement quickly.
5. There is a lack of flexibility. This is due to the structure of the model where a prototype has to be delivered before the final product and it follows this process.

2.2.4 Incremental model

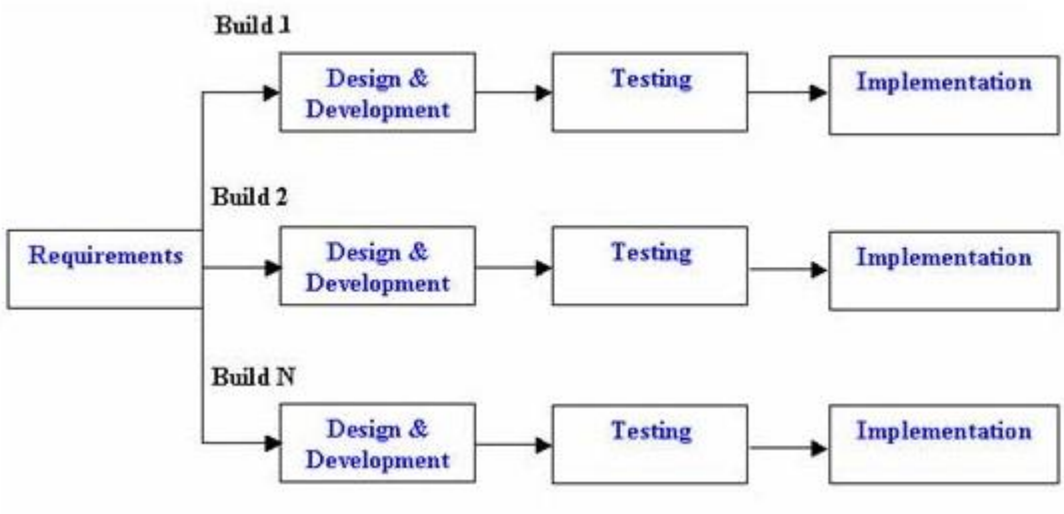

Figure 2.4 Incremental model diagram showing versions (http://tryqa.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/), (2008)

This model has the requirements broken down into increments which are produced from many development cycles. The cycles are separated into smaller cycles with modules that are easier to maintain. The modules go through requirements analysis, design, and development, testing, and implementation as shown in Figure 2.4. A working model of the software is produced as the initial module with new functionality added until the software is completed (Stoica et al., 2013).

Using the waterfall model or a rapid prototype model, an increment is created, and, based on the prior increment, the next increment is developed (Yu, 2018). A software product can be developed which allows users to utilize some functionality, while the other functions will continue to be developed simultaneously (Yu, 2018). This model applies to software projects in which all the requirements are not known upfront and could have risks in the design (Yu, 2018).

There are instances whereby the requirements for software development are clearly outlined, but the overall project does not allow for a step-by-step process (Mujumdar, Masiwal & Chawan, 2012). It may be necessary to produce a reduced set of functions for a user and thereafter expand on the capabilities in subsequent releases (Mujumdar et al., 2012). These scenarios allow for a process model to be selected for creating the required software in increments (Pressman, 2005). This model brings together attributes of the waterfall model in an iterative manner as a deliverable is created after each cycle which is increments of the required software (Mujumdar et al., 2012). The base product is the first increment where the primary requirements are delivered and the secondary attributes are still to be delivered (Pressman, 2005).

Advantages of the incremental model are:

1. The project is divided into smaller stages (Mujumdar et al., 2012). Each phase produces a working version of the software which fulfills some of the customer's requirements (Stoica et al., 2013).
2. A functional model is created early, and this allows for noteworthy feedback (Mujumdar et al., 2012). These functional prototypes are given to the customer who can provide feedback throughout the development (Stoica et al., 2013).
3. This feedback from one stage caters to the specification of the next stage (Mujumdar et al., 2012)
4. Risks are handled at the point of identification during the increment (Stoica et al., 2013).
5. It is helpful when staff is not available (Mujumdar et al., 2012). This is possible as the software is developed in increments which requires a smaller subset of developers who are responsible for one increment.
6. It allows for adaptability as it allows for changes if required (Stoica et al., 2013). If changes are required, another increment can be produced.

Disadvantages of the incremental model are:

1. The users are required to play an active role (Mujumdar et al., 2012). The customer can see the prototypes and ask for more functionality to be added to the next increment.
2. This requires the time of the users and this contributes to the delay in the project (Mujumdar et al., 2012). Since feedback is necessary, this could be time-consuming as the customer has to review after each increment.
3. Coordination and communication skills are key factors (Mujumdar et al., 2012). These skills are important as the customer has to be able to communicate requirements and feedback clearly or it could cause delays.
4. There may be confusion caused by possible informal requests for each stage (Mujumdar et al., 2012). This could lead to difficulty in keeping track of changes.
5. This may affect the scope of the project (Mujumdar et al., 2012)

The incremental model may be ideal for software projects that require the primary functionality of software early on in the project. In agile projects, builds or software functionality are delivered gradually and the software is reviewed by the team or via presentations to the customer (Popli et al., 2013). The issue of the customers' inability to change requirements in the waterfall approach is covered by the agile approach which allows the customers to change their requirements. Furthermore, bugs are carried from one phase to another in the waterfall approach and agile handles this by discovering and fixing bugs earlier in the lifecycle (Popli et al., 2013). The prototype model is in contrast to agile software development in which the interactions with the customer are vital to understanding the requirements of the system and there is always some customer representation with the project team. This allows for quick customer feedback and requirements can be changed as necessary (Popli et al., 2013). This leads to the next section on agile software development methodology which accounts for some of the shortcomings of the previous models discussed.

2.3 AGILE METHODOLOGIES

The Unified Process (UP) which was developed by Grady Booch, James Rumbaugh, and Ivar Jacobson is a framework that is made up of minor and major events and disciplines. It is also called the Unified Software Development Process (USDP) which is driven by use-cases and follows an iterative, incremental approach (Alhir, 2005). The Rational Unified Process (RUP) was created by IBM as a product that defines processes. The RUP is an extension of the UP and gives the information on how to execute projects while using the UP, with the addition of templates, guidelines, and tool support (Alhir, 2005).

Agile Unified Process (AUP) is a version of the RUP in a simpler form. It uses the RUP but incorporates an approach to software development utilizing agile methods in a simplified manner which is easy to use. The four phases of AUP are Inception, Elaboration, Construction, and Transition (Ambler, 2006). The inception phase aims to determine the project's scope, architecture, and to obtain funding and acceptance by the stakeholders. With elaboration, the aim is to provide validation for the system's architecture. Construction aims to create software regularly, incrementally, and meeting the stakeholders' highest needs of priority. The transition has the aim to deploy the application into production after validation (Ambler, 2006). UP has been applied by practitioners in a more-agile or less-agile way and is not dependent on its definition, but on how it is applied (Alhir, 2005). This agile application of UP is referred to as AUP.

Popli et al. (2013) shared that agile software methodologies emerged as a response to the traditional methodologies of developing software which entailed heavyweight activities with a focus on documentation. Traditional methodologies were not able to adapt to changing requirements and this gave rise to methodologies that are based on iterative development which were first realized in 1975 and became later known as agile methodologies (Awad, 2005). The name "agile" was coined in 2001 when a group of seventeen methodologists had a meeting on the software development methodologies and the common trends that were noted. The conclusion was that their methodologies were similar, and they named the processes as agile to denote them being light and appropriate. After this meeting, the "Agile Alliance" and the agile manifesto were created for agile software development (Awad, 2005). This agile manifesto was brought into existence in 2001, while agile as a methodology was derived in the 1990s (Mbelli & Hira, 2016). The reason for the formulation of this methodology was to show methods of producing software in a better manner that could fulfill the customer's needs in an environment that was iterative and controlled (Mbelli & Hira, 2016). This research by Mbelli and Hira (2016), showed that Scrum was the most popular agile methodology being used. The types of agile methodologies being followed are extreme project management, scrum, adaptive project management, and dynamic project management method and the most popular is scrum (Cervone, 2011).

According to Mushtaq and Qureshi (2012), the software industry favours agile methodologies as they efficiently produce quality software. Scrum and Extreme Programming are the most widely used agile methodologies (Mustaq & Qureshi, 2012), The discussion on Extreme Programming and Scrum follows.

2.3.1 Extreme Programming

This agile methodology is one of the most used and popular agile methods (Hameed et al., 2016). It entails being derived from a set of methods like customer collocation, customer satisfaction, and pair programming (Awad, 2005). XP was created by Beck Kent in 1999 to adjust the limit of the expected behaviour of the system and to dismiss most of the safety mechanisms that are exhibited by other methodologies (Tumbas & Matković, 2006). The software is created in iterations via sprints which are two weeks long. The user stories are attributes of the software that are developed by the designers and users (Tumbas & Matković, 2006). Extreme Programming (XP) places value in individuals as being the most vital factor during the software development process (Lippert, Wolf & Roock, 2002). The customer is required to be on-site and is a recognizable attribute of this method. As shown by Steinberg and Plamer (2003), XP emphasizes the agile methods of collaboration, the usage of support methods, and efficient and rapid development (Beck, 1999). XP relies on actual programming methods while scrum relies on management and organisation methods. XP is based on the ability to bring all the development team members to collaborate (Alfaki, Ali, Babiker, Ibrahim, 2016). Alfaki et al. (2016) emphasized that the important attributes of XP are the small iterations with small releases, quick feedback, customer involvement, coordination, communication, refactoring, continuous integration, and testing, joined ownership of code, and pair programming. XP, therefore, focuses on software development success irrespective of the changes in requirements.

XP was derived from the issues that were experienced as a result of the long cycles that were involved in traditional methodologies (Abrahamson, Salo, Ronkainen, & Warsta, 2002). XP comprises practices that have been used before, but XP has taken practices from older methodologies and uses them in conjunction with each other to create a new development methodology (Abrahamsson et al., 2002). The word “extreme” is derived from using the older practices and taking them to extreme heights (Abrahamsson et al., 2002). XP endeavours to produce successful software regardless of the requirements that change frequently in small-sized and medium teams (Abrahamsson et al., 2002). An advantage of XP is that it succeeded in environments that were completely different like organisations that had levels of control to those who had central or no control at all (Dyba & Dingsøyr, 2008). Trust and interpersonal abilities are vital attributes of an XP team (Dyba & Dingsøyr, 2008). Figure 2.6 shows the XP release cycle which starts with evaluating the system and follows with selecting the applicable user stories for the release in question. User stories are related to each release and the stories are broken down into tasks. The release is planned, the software is developed, tested, and then released.

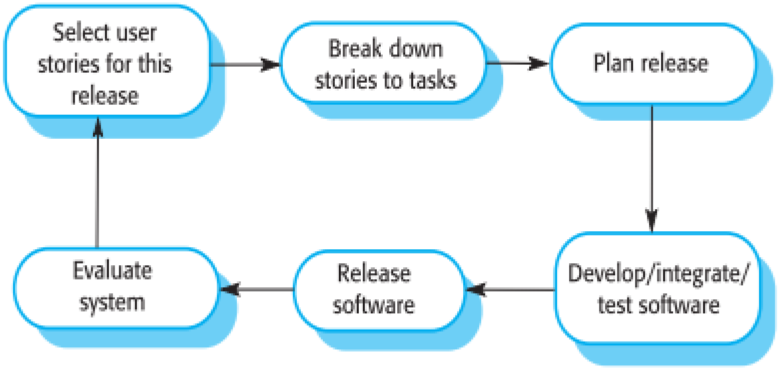

Figure 2.5 The XP Release Cycle (Munassar & Govardhan, 2010)

2.3.2 Scrum

The most popular and vastly used agile software development methodology is Scrum (Anwer, Aftab, Shah & Waheed, 2017). The scrum methodology is a method of software development that entails an agile manner that is controlled in an environment that is constantly in change (Cervone, 2011). It is similar to the scrum in rugby as the scrum occurs after a disruption and this can assist in managing the differences that could arise from different requirements in the team (Cervone, 2011). Scrum can be used to develop any functionality and is an iterative and incremental process. It delivers possible functionality after every iteration (Javanmard & Alian, 2015).

Scrum starts with the analysis of requirements in which the user stories are derived and are noted in the product backlog (Linke, 2019). The Scrum methodology defines three roles and four meetings. The roles outlined are the Scrum Master, Product Owner, and the Development team. Figure 2.5 shows the roles of the scrum team. The meetings outlined are the Sprint Planning, Daily Scrum, the Scrum Review, and the Scrum Retrospective (Linke, 2019). In the

Weekly Scrum, the next set of goals for the sprint is analyzed with regards to the requirements or the user stories and the estimation of the effort. The method aims to avoid a backlog within a sprint or that deliverables are not met as a result of the effort being understated (Linke, 2019). The Scrum Master and the Product Owner have the main control in a Scrum project. The Product Owner is recognized as being accountable for the product with the customer insight, while the Scrum Master is recognized as the owner of the process and solver of problems (Linke, 2019).

The Product Backlog is made up of a list of the attributes and amendments that have to be implemented by either the project team members, customers, or other departments (Awad, 2005). The Product Backlog is managed by the Product Owner (Awad, 2005). Sprints are the processes that should produce an increment of software that can be delivered to the customer. The sprints are usually thirty days long and allow for adhering to the environmental variables that can change like requirements, resources, time, knowledge, and technology. The project team has operational tools which are the Sprint Planning Meeting, Sprint Backlog, and Daily Scrum meetings (Awad, 2005). The Sprint Planning meeting is initially for the users, product owner, management, scrum team, and customers in which a specific list of functionalities that are agreed upon and then the Scrum Master and Scrum team concentrate on how the functionality will be delivered in the Sprint (Awad, 2005). The Sprint Backlog is the breakdown of attributes that are allocated to a specific Sprint and after all the attributes are developed, an iteration of the system is implemented (Awad, 2005). The Daily Scrum is held daily for about fifteen minutes to discuss the Scrum Team's progress and to understand any impediments that the team may be experiencing (Awad, 2005).

Scrum supports the people factor as the daily scrum meetings bring to light any impediments that the developers may be experiencing that could contribute to the backlog. Scrum also supports the process factor as the scrum meetings follow a specified process of attending to the backlog and planning each sprint. Since scrum involves the customer via the product owner's input, the people factor is lightly considered (Linke, 2019).

Scrum is successful and fits projects where requirements are not defined completely upfront and adaptability is expected during the software development life cycle (Rising & Janoff, 2000).

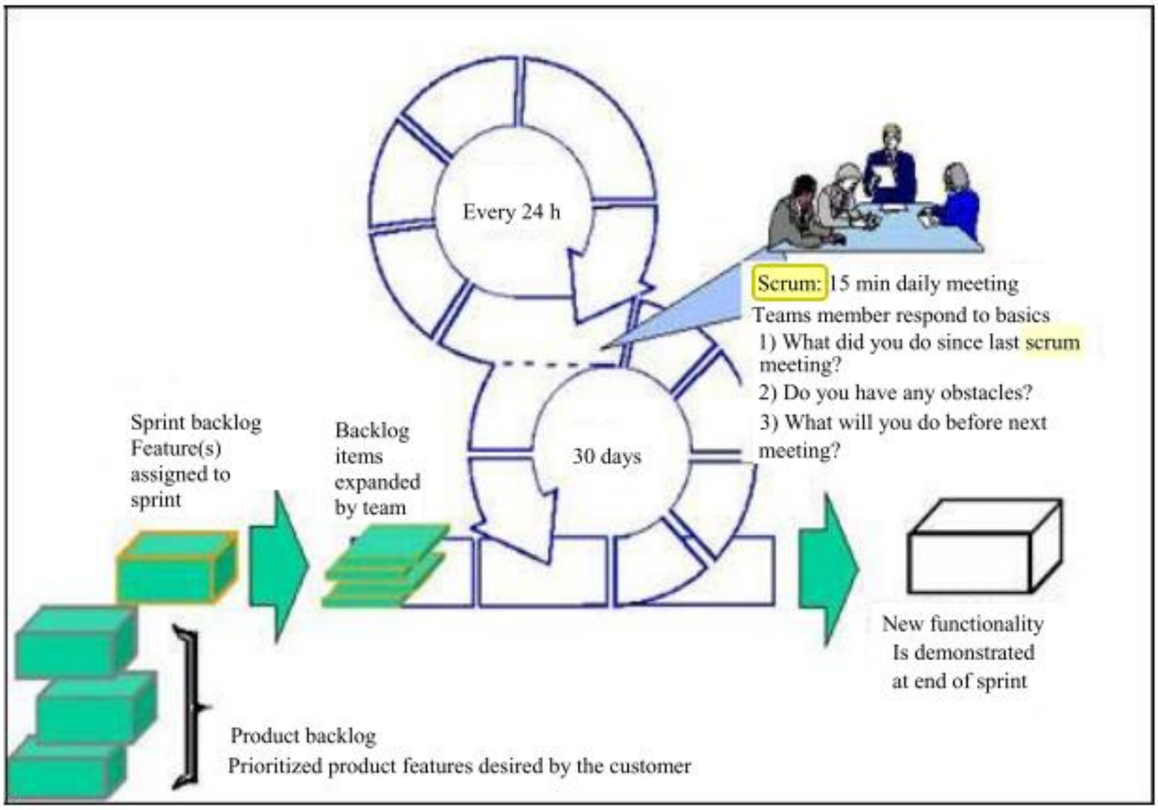

Figure 2.6 The Scrum life cycle (Madadipouya, 2015)

2.3.3 Scaled Agile Framework (SAFe)

Dean Leffingwell developed the Scaled Agile Framework (SAFe) in 2011 (Uludag, Kleehaus, Xu & Matthes, 2017). The Scaled Agile Framework (SAFe) is a framework that was established to manage large agile teams that comprise an excess of 50 developers who are situated in different geographical regions (Uludag et al., 2017). Agile, as a methodology increases in popularity but organisations are concerned with how to scale these methods as agile methods like scrum are not able to manage such a large number of developers (Uludag et al., 2017).

SAFe's objective is to provide a mechanism for the integration between agile methodologies that are Scrum, Extreme Programming (XP), Lean, and Product Development Flow (Turetken, Stojanov & Trienekens, 2016). According to Maarit (2014), scaled-agile frameworks are groups of conditions or rules that are used to change an organisation so that it displays the attributes that are related to an agile approach. SAFe is made up of three base levels which are

Portfolio level, Program level, and Team level with an extra Value Stream level for large organisations (Kalenda, Hyna & Rossi, 2018). SAFe is not about following a single methodology, but rather using the best practices for teams that suit them the best and will deliver successful projects. The foundation level exists across all the levels with additional attributes that provide organisational support (Kalenda et al., 2018). The Foundation layer is made up of the SAFe core values, the lean-agile mindset, and SAFe principles. The Portfolio level guides the organisation in its mission. It defines and oversees the core strategic decisions that will add value or benefit to the organisation (Kalenda et al., 2018). The Program level is responsible for implementing the mission and managing, supporting, and synchronizing the various agile teams that will produce the result. The lowest level of SAFe is known as the team level.

There are 6 squads for complicated solutions that need to be broken down into smaller parts and there is a product owner that manages each team (Kalenda et al., 2018). According to Uludag et al. (2017), each team is made up of five to nine members with a Product Owner and a scrum master. The Agile Release Train (ART) is shown in Figure 2.7 which also shows the full view of the SAFe. An Agile Release Train is a team made up of other agile teams that produce incremental functionality in the form of releases. SAFe is a framework to guide agile methodologies but is not a methodology itself and there are other scaled agile models (Uludag et al., 2017).

An example of another scaled agile model called the Spotify model was created in 2012 by Kniberg and Ivarsson which evolved from Spotify which is an audio streaming service (Cruth, 2020). The Spotify model became popular after it was created and focused on the organisation of work instead of adhering to guidelines or practices as is the case with SAFe (Cruth, 2020). The difference with the Spotify model is that it concentrates on how an organisation can be structured to ensure agility, whereas with SAFe, there are practices that govern how the framework is followed (Cruth, 2020). This model allows each team to choose their framework example, Scrum or Kanban (Cruth, 2020). Spotify is also an agile model at scale like SAFe, however, it is lightweight and focuses on the need for interactions to prevent working in silos (Paquet, 2019). It does not govern how the team should work in detail, while SAFe is heavyweight and reliant on practices and structure that the organisation should adopt (Paquet, 2019).

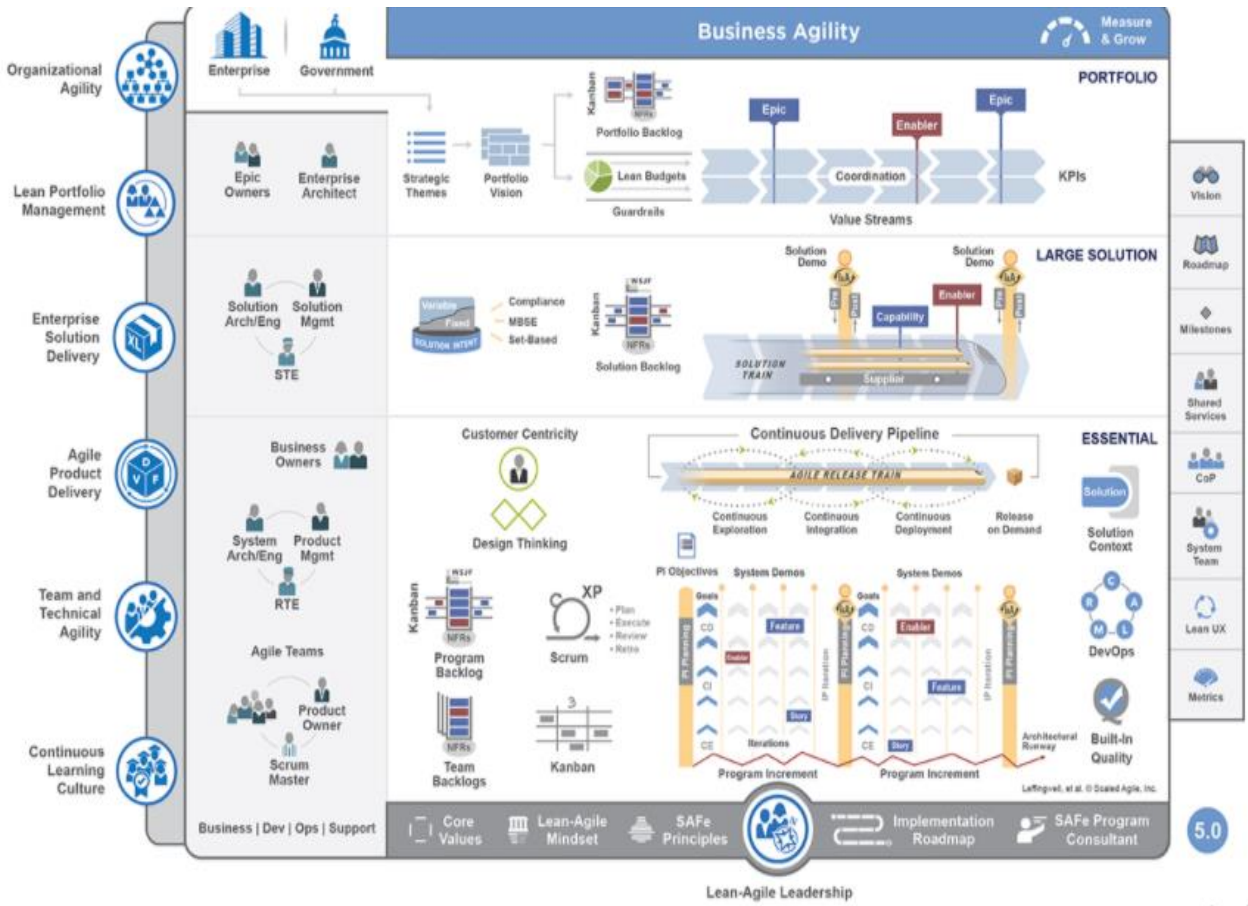

Figure 2.7 Scaled Agile Framework (*Scaled Agile Framework – SAFe for Lean Enterprises*, 2017)

2.4 AGILE METHODOLOGIES VERSUS TRADITIONAL METHODOLOGIES

According to Unhelkar (2013), an agile approach is ideal for small projects which are made up of 5 programmers and with a duration of about 6 months. Further, he stated that a large project with many teams involved in separate functionalities of the software at the same time and is most likely to be a Waterfall project. Almeida (2017) stated there are many positive attributes that agile methodologies bring to a project, but there is a reluctance to using it on a big scale by managers and engineers. Traditional methods have phases whereas agile methods produce iterations. Traditional methods are reliant on documentation, whereas agile methods have quick releases, short iterations, basic design, refactoring continuous integration, and knowledge of the team (Devi, 2013). Agile methodologies were introduced to improve customer collaboration and turnaround times that were associated with traditional methodologies like the waterfall methodology in which a deliverable was only possible at the end of the entire project.

Software development involving traditional methodologies is recognized as rigid and not able to adapt to change which is required by software development (Nerur, Mahapatra, &

Mangalaraj, 2005). For this reason, agile methodologies were introduced to overcome the challenges with traditional methods, and this proved to be a difficult transition (Tanner & von Willingh, 2014). This change from waterfall to agile methodologies is challenging as with agile methods, the outcome cannot be predicted, and it is not sequential from start to finish (Cohn, 2002). Teams using agile methods approach issues in different ways and the solution is chosen according to what is required (Kropp & Meier, 2015).

Previous studies show that traditional methodologies are mechanical to use (Abrahamsson et al, 2002). Agile development concentrates on the people aspect of development and it is this which gives agile methods the edge because people or human elements cannot be replicated like cost or technology (Pfeffer, 1998). The major distinction between agile and traditional methodologies is the adaptability characteristic (Javanmard & Alian, 2015). With the agile methodology, the team does not stop the work procedure if there is a change that is needed but establishes how to cater to the changes that arise during the project (Javanmard & Alian, 2015). Tables 2.1 highlights the comparison between traditional and agile methodologies from two sources that were combined to show the main differences that were evident. SAFe introduces another interface to manage the dependency between the agile team and the environment. The interface that exists between the development team and operations is called DevOps and is noted as the agile product delivery competency that is utilized in ART and solution trains (Theobald & Schmitt, 2020). This assists in agile scalability. DevOps is a way of bridging the gap between development and operations teams and can be used for further collaboration between the agile team and other teams. This improves the communication and scalability between teams in agile projects (Theobald & Schmitt, 2020).

Organisations that adopt SAFe generally have a scrum foundation, sponsorship from the executive level, and an inherent need for change (Piikkila, 2020). In the research by Kalenda, Hyna, and Rossi (2018), four key success factors influenced the organisation's adherence to agile. The factors were executive sponsorship, management support, unification of views and values, company culture, and existing Agile and Lean experience (Kalenda et.al, 2018). SAFe provides an option to scale agile to achieve the business needs of the organisation (Piikkila, 2021). Therefore, SAFe is successful in scaling Agile.

Table 2.1 The comparison between Agile and Traditional Methodologies (Javanmard & Alian, 2015 and Leau et al., 2012)

	Agile	**Traditional/Heavy Methods**
Team Size	Small/Creative	Large
Project Size	Small	Large
Upfront planning	Minimal	Comprehensive
Requirements Analysis	Iterative approach	Detailed requirements profile
Approach	Change Adaptability	Predictive/Change Sustainability
Development direction	Can be changed at any time	Fixed
Additional quality required for developers	Interpersonal skills/Business knowledge	Not specifically required
Modification costs	Low	High
Customer Interaction	High	Low
Testing	After each iteration	When the development phase has been completed
Management style	Decentralized	Autocratic
Perspective to Change	Change Adaptability	Change Sustainability
Documentation	Low	Heavy
Emphasis	People-Oriented	Process-Oriented
Cycles	Numerous	Limited
Culture	Leadership-Collaboration	Command-Control
Success Measurement	Business Value	Confirmation to plan
Return on Investment	Early in Project	End of Project
Domain	Unpredictable/Explorable	Predictable

The Standish Group was created in 1975 and is a research advisory organisation with a core focus on the performance of software development projects around the world (About The Standish Group - The Standish Group, 2019). According to Mersino (2018), The Standish CHAOS reports are released every two years since 1994 to show the number of project failures and successes in the global development of software. Figure 2.8 shows that agile projects are two times more likely to be successful than waterfall projects and waterfall projects are two-thirds more likely to fail than agile projects (Mersino, 2018).

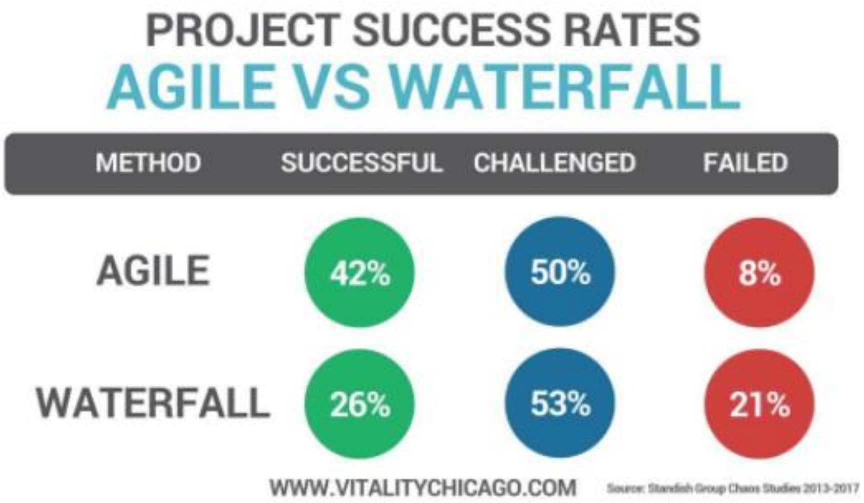

Figure 2.8 Failure and success of agile and waterfall projects from 2013-2017 (https://vitalitychicago.com/blog/agile-projects-are-more-successful-traditional-projects/)

2.5 CRITICAL SUCCESS FACTORS

The Standish Group's (1994) CHAOS report indicates that the top five success factors for projects are user involvement, top management support, clear requirements, proper planning, and realistic expectations (Mersino, 2018). As can be seen from Figures 4 to 6, the failure rate of waterfall projects is three times that of agile projects. Agile projects have been consistently twice as successful as waterfall projects. Critical success factors (CSF) were developed as a means of assisting team managers to ascertain which information is critical to them to reach their goals (Rockart, 1979). Critical success factors are considered the focus points on which the organisation should work to perform well regularly to reach its objectives (Gates, 2010). Sudhakar (2012) identified CSFs as those variables that contribute to the successful outcome of the agile project. Managers are aware of these focus areas when they outline organisational objectives and they should ensure that their activities are in line with reaching these objectives (Caralli, 2004). Close communication, the collaboration between the programmers and the clients, and small, focused teams are the central focus point of agile methods (Jalali & Wohlin, 2010). People are an important factor to keep in mind as they are one of the most important resources and can pose a challenge to agile methods (Noruwana & Tanner, 2012).

Projects are successful when the critical factors are properly understood and failure allows for the issues to be eliminated in the future (Hameed et al., 2016). According to the study by Garousi et al. (2018), three CSFs have the strongest relationship with project success. These are a) the team's skill or experience in the software methodologies, b) the team's skills to implement tasks, and c) the monitoring and maintenance of the project. Chow & Cao (2008) recognized three critical success factors that affect projects using agile which are delivery strategy, agile software engineering techniques, and team capability. The study entailed using quantitative methods to investigate projects to identify the critical success factors in agile projects. The survey data was retrieved from 109 agile projects that belonged to different organisations, industries, locations, and sizes. The literature discussed a high number of contributors that influence agile projects, but only a few key success factors were found. Good agile management processes, an agile-friendly team environment, and strong involvement of the customer are three other factors that were identified to be critical to certain success areas (Chow & Cao, 2008).

Later, Vithana, Fernando & Kapurubandara, (2015), investigated the view of Sri Lankan software practitioners on the organisational and people factors that affect the success of agile software projects. Based on the results, technical competency from the people factors had the

highest influence on the agile project success (Vithana et al., 2015). The study by Chow and Cao (2008) attempted to reduce the CSFs to only three critical success factors. They believed if the project had a high-caliber team, followed rigorous Agile engineering, and implemented an appropriate Agile-style delivery strategy, the project would likely succeed.

The other factors of communication and negotiation, societal culture, personal characteristics, training, and learning do not substantially affect the agile project's success. The organisation factors, team size, customer satisfaction, decision time, planning, and corporate culture affect agile project success (Vithana et al., 2015). Customer commitment, customer collaboration, team distribution, and control do not affect the agile project's success to a high degree (Vithana et al., 2015). This may be because in this study most of the customers were offshore, so customer collaboration was not possible throughout the project (Vithana et al., 2015).

The organisation's critical success factors are key aspects that are specific to an organisation and the organisation must be successful in these areas to reach its goals (Caralli, 2004). The success rate of software projects can increase if organisations focus and endeavour to work on the CSFs that have been identified (Sudhakar, 2012). A high-performing team, proper agile software practices, and an appropriate agile delivery could lead to a successful project (Chow & Cao, 2008). According to Kaur and Singh (2016), the organisation can be productive by withstanding the challenges and concentrating on the success factors while adapting to agile methods. In a study by Lam, Cheung, Wong, and Chan (2013), the critical success factors are a good project management process, a project definition process that is clear, strong customer interaction with the customer, and strong commitment from management. In the study by Kulathunga and Ratiyala (2018), critical factors influencing the success of scrum projects were identified as organisational, people, process, technical, and project factors which are the same five factors that were investigated in this research with the inclusion of test automation and cloud computing.

The following factors were recognized as agile critical success factors from the research discussed above.

2.5.1 Organisational Factors

The team in a project refers to all the members who are a direct part of the development like the project manager, product owners, scrum master, developers, and testers and they all follow a common methodology (Kropp & Meier, 2015). The organisation is an extension of the team and is composed of the team and other individuals who are related to the project like the clients, users, and project sponsors who work together, while the company is the corporation or enterprise in which the development occurs (Kropp & Meier, 2015).

If an organisation decides to implement agile, the culture of the organisation and the thinking have to be changed as agile is a method of working and delivering and this is challenging to change (Hameed et al., 2016). Successful agile teams have a culture that is open and transparent, and they work hard to achieve quality work (Kropp & Meier, 2015).

In the study by Vithana et al. (2015), it was evident that the organisational factors of decision time, size of the team, corporate culture, customer satisfaction, and planning contribute to the agile project success. It is shown in this study that the distribution of the team and control, the commitment of the customer, and customer collaboration do not significantly affect the success of the agile project. This could be evident as most of the customers for Sri Lankan projects are offshore and therefore, collaboration with the customer and commitment during the software development is not possible with these customers (Vithana et al., 2015). This shows that the type of customer and the location of the customer could influence how the organisational factors affect the agile project success.

Organisations that want to adopt agile methodologies need to have a culture of quick communication, placing trust in people, encouraging the decision-making of developers, facilitating changing requirements, and spurring efficient feedback from customers (Lindvall et al., 2002). This shows that the culture of the organisation influences agile project success and for this success, the culture has to be agile-centric which entails the characteristics of agile projects that thrive on efficient communication, rapid delivery, customer involvement, and teamwork.

If the organisation's culture cannot encourage and promote agile, then the organisation cannot be called agile as agile entails having a culture that is agile-oriented (Lindvall et al., 2002). Organisations that adopt agile methodologies require agile-oriented thinking and the organisational culture has to transition as agile is a method of working that embraces the different characteristics of the project team (Hameed et al, 2016).

2.5.2 People Factors

People factors in a team are stable team members, team members who are dedicated, and T-shaped individuals, and this is linked to the members' abilities and the environment that they work in (Totten, 2017). The idea of a T-shaped individual means that the individual has extensive expertise in one aspect and a good understanding of the other aspects which allows the team to get together to finish allocated work (Demirkan & Spohrer, 2015). The people factors are important in the development and can involve the education element which entails that the project team members should acquire agile skills to implement them and to be able to assure the team, while the project manager should be skilled in coordinating the changes (Hameed et al., 2016). Both the project manager and the project team members should be vested in their roles and responsibilities within the project (Hameed et al., 2016). Training and education positively influence the projects that are changing their processes to be more agile (Wan & Wang, 2010). This indicates that individuals who are adequately trained could contribute to the success of agile implementations.

The project team requires members who are competent and experienced agile practitioners (Islam, 2016). Team members who are motivated will be able to satisfy client requirements by developing quality deliverables as they will provide the team with proper decisions and error findings (Islam, 2016). Creating a clear line of communication between the project team and the client influences agile project success (Islam, 2016). The positive outcome of agile projects relies upon the relationship between individuals and resources (Misra, 2007). The reason for this is that agile projects focus on individuals and interactions, collaborating, and working with the customer, and dealing with changes that are requested by the client (Misra, 2007). Competency is an important factor for project success as the scrum method strives to deliver operational software quickly and only a competent performing team would be able to deliver this (Kulathunga & Ratiyala, 2018). It is the duty of the manager of the project to choose the right team members with the right attitude and acumen for the team (Kulathunga & Ratiyala, 2018). The project team's capabilities, experience, and expertise would influence the agile project success. According to Garousi et al. (2018), the three major project success measures are team building and team dynamics, team satisfaction, and higher management satisfaction (Garousi et al., 2018). The team building and dynamics aspect encompass reliance and trust within the team. Team satisfaction involves the team members being happy to work in the team and top management satisfaction entails management being happy with the performance of the project team. The team of project members can only improve the likelihood of success if they concentrate on all critical success factors (Garousi et al., 2018). If there is no encouragement from top management, then software process improvement cannot survive for a long period (Paulk, 1997). Organisations that have the agile capability should receive the support of the organisation (Zhang & Sharifi, 2000). This means that organisational support positively influences the success of the implementation of agile process improvement. If the organisation is striving for agile project success, it must utilize proper techniques and tools for this purpose (Zhang & Sharifi, 2000). The organisation's support and culture filter through to all the critical success factors. Education and training are instrumental in positively influencing the successful implementation of agile process improvement (Wan & Wang, 2010). The agile culture must be embedded with agile methods which means that mutual trust and culture must exist within the corporate culture (Wan & Wang, 2010).

2.5.3 Process Factors

Agile teams align their processes to suit the changing nature of the agile project (Kropp & Meier, 2015). The members of the team believe that they can govern the change in processes because of the agile nature of a project (Kropp & Meier, 2015). Adopting processes of a client-oriented approach to the project with meetings regularly can influence the project to attain its goals from the beginning of the project (Islam, 2016). Project management processes significantly influence the success of software development projects (Ceschi, Sillitti, Succi, & De Panfilis, 2005). Using project management processes in agile methodologies results in higher success and therefore also increases client satisfaction (Buresh, 2008).

A structured project management methodology contributes to the success of agile projects. There are many agile methods to choose from such as scrum, extreme programming (XP), crystal, and more, but it is vital to select the appropriate method for each project and to note the mechanism to integrate with external processes (Abdalhamid & Mishra, 2017).

2.5.4 Project

Agile methods can manage changing requirements, they are most suitable to projects where the requirements can change and are not strictly defined (Hameed et al., 2016). The type of project and its requirements could influence the success of agile projects. The type of project can be life-critical or non-life critical (Kulathunga and Ratiyala, 2018). Agile processes have been used for projects that were not critical to life to make sure that these projects are a success (Chow & Cao, 2008). It is known that agile methods are most appropriate to projects where requirements are not well characterized and fluid because they look for containing change without difficulty (Abdalhamid & Mishra, 2017).

Traditional methodologies follow a fixed scope whereas the scrum methodology follows a variable project scope. Since there are new requirements that can arise within the scrum methodology, the scope of the project can be affected (Joseph, Marnewick & Santana, 2016). Projects that have a project scope that can be changed on demand are appropriate for agile projects (Chow & Cao, 2008). The schedule of the project can change since the scope of the project can change in the Scrum method (Kulathunga and Ratiyala, 2018).

It is imperative to ensure that the scope and goals of the project are defined at the outset and signed off with the client before the project is implemented (Chow & Cao, 2008). Based on the organisational requirements, the development team decides what is to be delivered in each sprint (Cervone, 2011). The product backlog is created by all the clients' requirements and sprint planning entails the breakdown of the backlog into sprint backlog items. This shows that the well-defined scope of the entire project is critical for the success of the project.

2.5.5 Technical Factors

To maximize agile benefits, organisations should use appropriate tools and technology (Zhang & Sharifi, 2000). These tools and techniques that are used should support agile methods and approaches (Tanner & von Willingh, 2014).

In a study by Hameed et al. (2016), the organisation was able to produce the customer requirements quickly and in an efficient manner with the use of technical practices like the development that is test-driven, refactoring, pair programming, continuous integration, and collective ownership.

Requirement changes are more than likely in agile projects and the change is anticipated (Misra et al., 2009). Manuals on the procedures and the system provide the users with the technical aspects of the product and the solution that is being produced (Misra et al., 2009). The tools that support the development will assist the project team in delivering their tasks with ease (Nasir & Sahibuddin, 2011). In the study by Chow and Cao (2008), the results indicated that the technical dimension which included agile software engineering techniques and delivery strategy was the most critical in affecting the agile project success as it was based on all four success attributes of quality, scope, time and cost. Islam (2016), also suggested that technical factors are the most influential factors to the success of an agile project, followed by process factors, organisational factors, people factors, and project factors.

2.5.6 Test Automation

Test automation was separated from technical factors as the technical factors covered the tools that were required for the success of agile projects. Test automation covered all automated testing aspects of the agile project. Although automation is part of agile technical factors, the test automation factor as a CSF in this research concentrated only on the automation of testing elements, while technical factors covered tools, techniques, and processes. This was to allow for the understanding of the tools and techniques that were "technical," but not necessarily "automated."

Software testing is important for the verification and validation of software (Kumar & Mishra, 2016). It is important as it ensures the quality of the software and testing accounts for almost 60% of the total cost of software (Kumar & Mishra, 2016). With agile testing, the team is not limited to finding errors, but also to avoid failures (Collins & de Lucena, 2012). Due to the continuous development of functionality in an agile process, testing is frequent. If testing is carried out manually the ability to rapidly create these new functionalities can be compromised. Therefore, it is important that testing is automated (Collins & de Lucena, 2012). Dustin, Rashka, and Paul (2008) defines software test automation as the usage of testing tools that are automated to automate testing activities such as test script development and execution and testing requirements validation. This automation will increase the efficiency of repetitive steps especially during regression testing whereby there is an incremental and iterative execution of test cases after software changes (Karhu, Repo, Taipale, Smolander, 2009). In the study by Collins & de Lucena (2012), the unity of the project team ensured cooperation, knowledge was shared between testers and developers and the test automation environment was the responsibility of the project team. This indicates how test automation is not limited to software testers, but the whole agile team. Team collaboration was indicated as an important factor in agile project success in research in the agile field and this was also evident for test automation (Collins & de Lucena, 2012). The research by Kumar & Mishra (2016) on the impacts of test automation showed that the software quality improved with test automation as there were fewer errors with automated test cases than with manual testing. It was also observed that the time and cost aspects were positively impacted. The time in testing was reduced by test automation and the availability increased in all cases. The maintenance and implementation costs of automated test cases are high, but the return on this investment is worthwhile when automated tests are rerun many times (Kumar & Mishra, 2016). Test automation improves the quality of software and positively influences the effectiveness of the testing (Kumar & Mishra, 2016).

2.5.7 Cloud

Agile development lacked a development platform that could support quick development cycles (Kalem, Donko & Boskovic, 2013). Cloud computing and agile methodologies are appropriately matched to one another i.e., the perfect environment for development in agile is cloud computing (Sayeed, Hassan & Muttoo, 2017). It allows you to deliver important functionality to your clients quickly, gathers an immediate response, and can implement quick amendments as a result of the quick feedback (Sayeed et al., 2017). Cloud can positively impact agile development in terms of the positive characteristics of development in agile, development infrastructure, and collaboration (Younas et al., 2018). Development in agile can be assisted and accelerated by the services and quality characteristics of cloud computing, which can assist in scalability, interoperability, and maintainability within agile software development (Franken, Kolvenbach, Prinz, Alvertis, Koussouris, 2015). Cloud computing positively impacts the development infrastructure by providing integrated development environments (IDEs), automated testing, and tool support. Team and user collaboration are also enhanced in the cloud environment (Younas et al., 2018). The duration of phases in agile development with cloud computing is shorter as cloud computing allows for quick resource allocation and easy change management. There is better communication with cloud tools example, Skype which can be used for meetings when the customer is not available to attend in person (Kalem et al., 2013). This will eliminate the room for failure that results due to a lack of communication in projects. Cloud computing allows for easy collaboration and customers can access the software from any location with internet access for testing purposes which reduces the testing time and the need to physically visit the team's physical location (Kalem et al., 2013). The key benefits of utilizing agile methodologies with cloud computing are the increased application quality, the proper usage of resources, the quicker time to deliver to the market, and the savings in cost (Kalem et al., 2013). Cloud computing provides a development environment that allows for sharing and this facilitates the online updating of software code as libraries are in a shared online platform (Almudarra & Qureshi, 2015). This version control is key in agile software development and is a benefit of cloud computing. GitHub which is one of the largest online platforms for hosting code, facilitates version control, issue tracking and allows for collaboration between developers in the user or team repositories which can be either public or private (Franken et al., 2015).

2.6 The Study's Conceptual Model

Based on the success factors discussed above and the objectives that the research intends on achieving, the conceptual framework for the research is depicted in Figure 2.9.

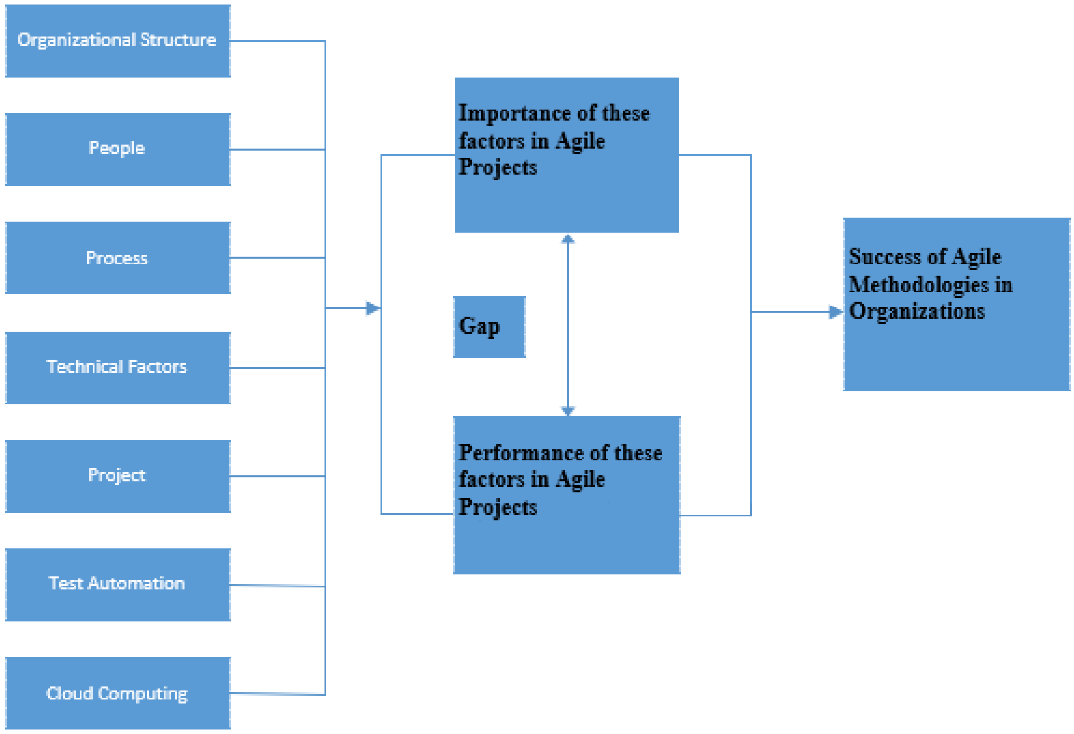

Figure 2.9 Comparative Ranking of the CSF for Agile Methodology

2.7 CONCLUSION

The literature on software development, traditional software methodologies, and agile methodologies were discussed in this chapter. Examples of traditional and agile methods were examined, together with advantages and disadvantages, and the comparison between the two methods was discussed. The critical success factors that affect the success of agile projects and the two additional factors of test automation and cloud computing were also discussed. The conceptual framework for the study was presented. The research methodology of the study is discussed in Chapter 3.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 INTRODUCTION

This chapter explains how the research was implemented and the research tools that the researcher used. The research methodology for this study is defined in this chapter. Research methodology describes the core mechanism of collecting the research data. The methodology choice, sampling strategies, sample size, and data collection methods are outlined. The research was planned and designed to ultimately answer the research questions of this study by collecting viable research data. The data collection method used in the study was selected by the research design.

3.2 RESEARCH DESIGN

The discussion of the research design and methodology to be followed in this study will be guided by the Saunders 'Research Onion' depicted in Figure 7 (Saunders, Lewis & Thornhill, 2019). The research designs and strategies are made up of the methodologies and guidelines that are used to implement the research (Hakansson, 2013). The research strategies and designs are the guidelines, or the methodologies, for carrying out the research.

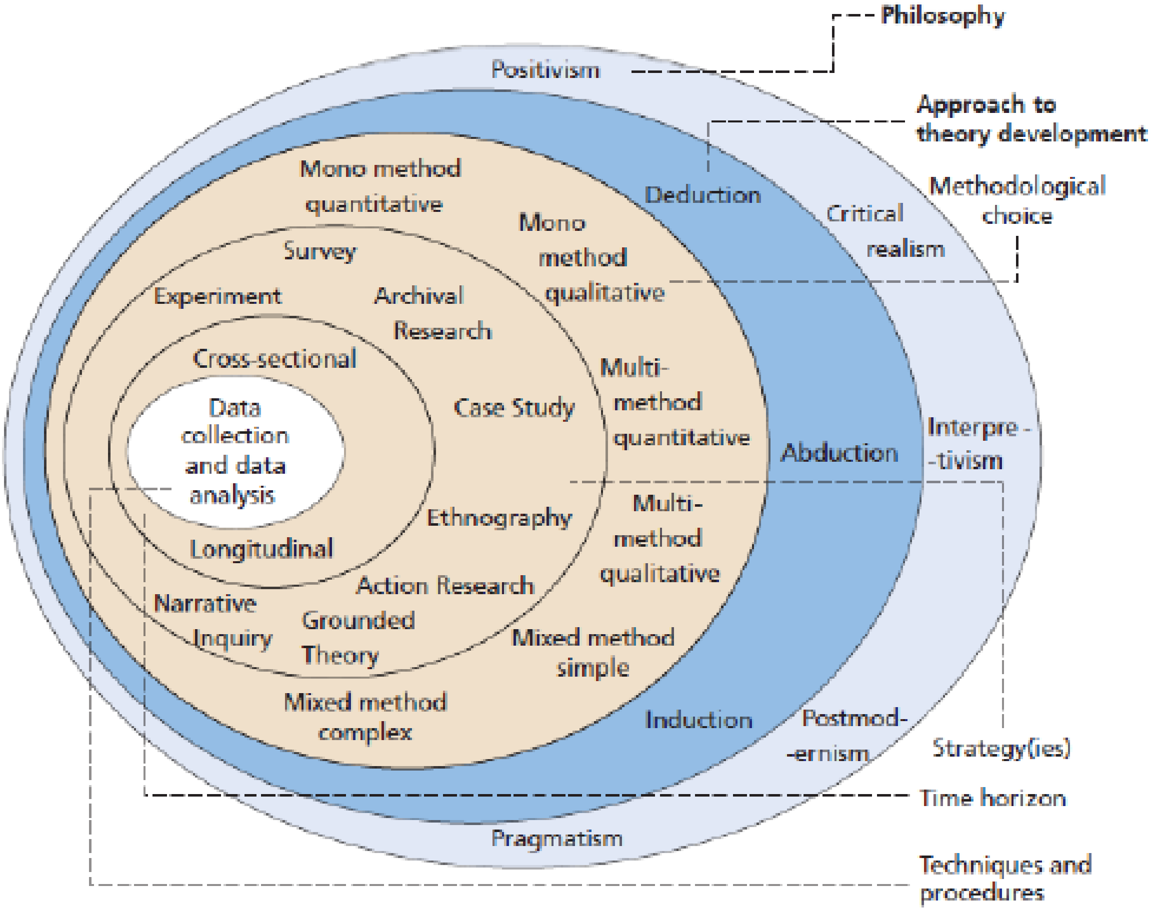

Figure 3. 1 The "Research Onion," (Saunders et al., 2019)

3.3 RESEARCH PHILOSOPHY/APPROACHES

Figure 3.1 dissects Saunder's Research Onion and shows the layers of research that resemble the layers of an onion and peel away to eventually culminate at the centre. The centre of the Research Onion is the Data Collection and Data Analysis. This research will use the positivist philosophy from the first layer, the Deduction approach from the second layer, the Mono method quantitative from the third layer which leads to the Survey choice from the fourth layer, and the Longitudinal approach from the fifth later. These five layers end with the centre being the Data Collection aspect which is covered in this chapter and the Data Analysis techniques which are covered in Chapter 4.

According to Saunders et al. (2019), the five major philosophies are positivism, critical realism, interpretivism, postmodernism, and pragmatism. Positivism entails the philosophical standing of the natural scientist which involves an end-result that is similar to that in the physical and natural sciences and that can be observed (Saunders et al., 2019). Critical realism entails what can be observed and felt concerning the reality that manifests in what can be seen (Saunders et al., 2019). Interpretivism is a subjectivist philosophy as it views humans as being different from physical events since humans attach meaning to events (Saunders et al., 2019). Interpretivism focuses on experienced events and includes their interpretations and the interpretations of their participants in the research (Saunders et al., 2019). Postmodernism focuses on questioning what has been accepted to seek alternative explanations that have not been given the chance to surface (Saunders et al., 2019). Pragmatist ontology, epistemology, and axiology are involved in improving practice and pragmatists use a range of research strategies that are governed by the research problems in question (Saunders et al., 2019). The researcher will utilize the positivism philosophy for this research study. The positivist assumption is that unbiased facts give rise to quality scientific proof that can influence the choice of quantitative research methods (Saunders et al., 2019). The researcher selected the positivist approach as this study aimed to be unbiased and to strengthen the validity of the data. The choice in this philosophy stems from the choice in the research method that will be discussed. Positivism focuses on measurable results and this is linked to the research method which is quantitative and will produce results that can be scientifically analyzed. The respondents' answers will provide the research data which will be translated into research findings via statistical methods. The findings will be scientifically examined via statistical testing and this ties to the positivist approach which is vested in natural science and observable results.

3.4 SURVEY DESIGN AND LAYOUT

The questionnaire was divided into two sections, namely Section A and Section B. Section A comprised of the demographic information of the respondents. Section B comprised of the research questions which included Likert scale questions from 1 to 5 which related to strongly disagree, disagree, neutral, agree, and strongly agree. The first set of 49 questions were the evaluation of the critical success factors that affect agile project success in terms of importance. The next set of 49 questions were the corresponding questions in terms of performance and the Likert scale for these questions was 1 to 5 which related to none, poor, fair, good, and very good. The importance questions required agreement or disagreement, while the performance questions required poor or good responses. Each importance question had a corresponding performance question. The questionnaire design aimed to be able to compare the perceived importance of the critical success factor with its perceived performance in agile projects. The last set of questions are to measure the respondents' perceptions of the success parameters of quality, scope, time, and cost that affect agile projects. These questions were independent of the importance and performance questions and were based on a Likert scale of strongly disagree to strongly agree to gauge what a successful project meant to the respondents.

3.5 STUDY SITE AND TARGET POPULATION

The population is the likely or possible participants that the study will focus on (Bacon-Shone, 2015). The target participants or population were software or information technology professionals working in an agile environment. The study site was South Africa. Questionnaires were distributed to the SGIS (Spar Group Information Services) department in Pinetown, Durban. The researcher then distributed most of the questionnaires to LinkedIn professionals who were targeted via purposive sampling. The objective was to obtain as many completed questionnaires as possible. This study targeted agile professionals in South Africa and was not limited to one organisation as LinkedIn is a professional platform with professionals from various organisations. Each potential LinkedIn respondent was purposively targeted via keyword searches. Each potential respondent was contacted directly via LinkedIn messaging and asked for their permission before the web link to the questionnaire was sent to them. The questionnaire was only sent to the individual upon their confirmation to participate. LinkedIn was used as a platform to source potential respondents, however, each respondent was personally contacted and given an overview of the research. The respondents were asked if they were willing to participate in the research and to validate that they did have experience or were working in agile projects. No data was sourced directly from LinkedIn. The data was sourced from the completed questionnaires only. Each potential respondent was contacted personally to confirm that he or she had experience in agile projects. Two conditions were required to be fulfilled before the questionnaire link was sent. The first condition was that the respondents had to be willing to participate in the research and the second condition was the respondents had to confirm their experience in agile projects. The questionnaire link was then sent to validated respondents. Some potential respondents confirmed that they did not have exposure and experience in agile projects and they were not sent the link to the questionnaire. This method was time-consuming, however, it allowed the researcher to validate that the responses were from practitioners that were involved in agile projects and confirmed that they were participating in the research without any coercion.

3.6 APPROACH TO THEORY DEVELOPMENT

Approach to theory development is vital due to three reasons which are: it allows the researcher to make a sound decision regarding the research design, it allows for the thinking on which strategies of research will be suitable, and the learning surrounding different traditions of research which will enable adapting the designs of research for accommodating conflicts (Easterby-Smith et al., 2012). Deduction, induction, and abduction are three central approaches to theory development (Saunders et al., 2019). Deduction entails the process where a theory and hypothesis are formulated and the hypothesis is tested by a research strategy that is designed (Saunders et al., 2019). Induction involves the process whereby data is collected, and the data is analyzed which leads to the formulation of the theory based on the results (Saunders et al., 2019). Abduction involves investigating a phenomenon with data collected, identifying scenarios and patterns, and formulating a new or changing theory that is tested with more data collection (Saunders et al., 2019). In this research, the central approach that was adopted was deduction as the study was based on a conceptual model that became the theory that supported the study.

This study used questionnaires that were designed specifically for the testing of the hypotheses and answering the research questions. This relates to the deductive approach where the hypothesis is tested using statistical techniques. The research questions were formulated, the hypotheses were developed, and the research strategy was designed via the questionnaire.

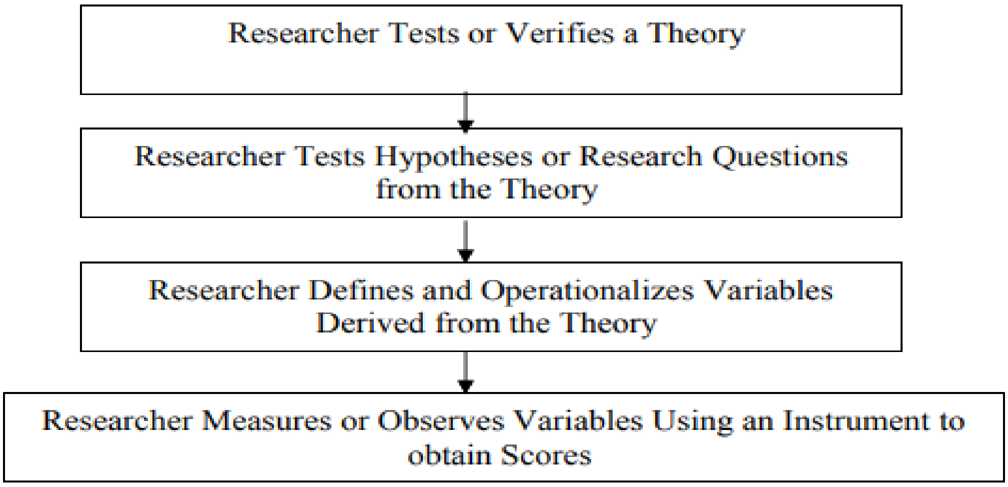

Figure 3.2 The Deductive Approach (Creswell, 2002)

As noted in Figure 3.2, the deductive approach which was adopted in this research follows the process whereby hypotheses are created, the research methods are selected and then used to test the hypotheses (Creswell, 2002). Further, it shows how the researcher starts with a theory, derives questions from the theory, derives variables from the theory, and measures findings via a research instrument which was the survey for this study.

3.7 METHODOLOGICAL CHOICE: (RESEARCH APPROACHES)

According to Creswell (2009), the two predominant types of methodological designs are quantitative and qualitative. There are several differences between qualitative and quantitative research methods with The differences between qualitative and quantitative research are highlighted in Johnson and Christensen (2008) as well as Lichtman (2006). An important difference is the type of data that is collected in each method. Qualitative methods collect data in the form of objects, words, or images while quantitative methods collect data in the form of numbers and statistics. Ahmad, Wasim, Irfan, Gogoi, Srivastava & Farheen (2019) stated that qualitative methods are exploratory, unstructured methods that collect data on complex topics that are not possible to attain with quantitative methods. It can create ideas for further research using quantitative methods. Qualitative design and research are defined as a mechanism for investigating and learning the way people derive an understanding of a problem or situation (Creswell, 2009). Non-numerical data are commonly collected using interviews, group discussions, and document analysis. This data is analyzed to generate trends that allow a researcher to interpret the meaning of the data.

Creswell (2009) describes quantitative research as a mechanism for validating objective theories by analyzing the association that exists among variables. This normally occurs with the collection of data on different variables and then uses statistical methods to analyze the relationship between these variables. According to Apuke (2017) quantitative research involves using techniques to collect and analyze numerical data to answer questions like what, when, how, who and how many. The study by Apuke (2017) also recognized that quantitative methods can be classified into survey research, experimental research, causal-comparative research, and correlational research. Ahmad et al. (2019) emphasize that quantitative research has a reliance on natural sciences from which numerical data and information are derived.

Quantitative research is also referred to as empirical research as it can be measured and aims to create a cause and effect relationship between two variables by using statistical, computational, and mathematical techniques (Ahmad et al., 2019).

The research method chosen for this study is the quantitative method, which is appropriate since this study will be using the factors that were already discovered in previous research to measure agile software practitioners' and agile team members' perceptions of these factors. This will allow for the attitudes of these agile professionals to be quantified enabling the researcher to establish the importance of these success factors that exist in an agile environment. This method also emphasizes the positivist paradigm and deductive approach highlighted above. According to Taylor and Medina (2013), the positivist paradigm is commonly used in graduate research studies to test hypotheses and aims to explore, support, and conclude law-like behaviour patterns.

The positivist paradigm which centres on unbiased results will be followed as the questionnaires are anonymous. This paradigm uses the quantitative method mainly, experimentation, control groups, and the checking of before and after tests to measure scores (Taylor & Medina, 2013). The deductive approach is based on arriving at conclusions from findings and this will be evident from the questionnaires.

This study uses the findings from the measurable input of the respondents hence quantitative research was applicable. The use of quantitative research for this study was applicable and suitable for the type of questions that were being asked. This was an extension of the positivist approach which led to this research strategy. Further, the factors that affect agile success have been documented in the literature and this study aims at quantifying the importance and performance of these factors.

3.8 DATA COLLECTION STRATEGY

This study will use primary data and survey research. Primary data is data that has been collected from a respondent's experience while secondary data is collected from an already published resource example, books, research articles, and newspapers (Kabir, 2016). According to Creswell (2009), survey research involves examining a sample from a population to derive the feelings or beliefs of the population via the results obtained. Closed-ended questionnaires were selected to facilitate quantitative research.

3.8.1 Sampling Strategies

The two types of techniques for sampling are probability sampling and non-probability sampling. Probability sampling is also known as random sampling while non-probability sampling is also referred to as non-random sampling. A sample where all the units of sampling have an identified non-zero probability of choice or selection is a probability sample. In contrast, Sedgwick (2013) adds that convenience sampling is a non-random sampling technique that entails recruiting participants to a study based on their availability and accessibility. On the other hand, random sampling is based on the premise that every member of the population has an equal probability of being selected for the research (Sedgwick, 2013). However, this study will not utilize a probability sample as the strategy because of the difficulty in obtaining a sampling frame. The non-probability technique, purposive sampling will be used. The software or agile practitioner population size is difficult to estimate, so individuals will be targeted for sampling due to the large population of software practitioners in South Africa. Purposive sampling is a method that incorporates the characteristics of the intended individuals into the choice of selecting them for the study (Etikan, Musa, Alkassim, 2017). In other words, the individuals who match a particular criterion will be targeted for the research. For this study, respondents must be working or worked on agile projects. Each potential respondent was contacted personally to validate whether he or she had experience in agile projects. The questionnaire link was only sent once this was confirmed as the survey required knowledge of agile and experience in agile projects.

Purposive sampling enabled the researcher to use judgment when selecting potential respondents and this saved time as the researcher was able to source targeted individuals via LinkedIn. Purposive Sampling and Convenience Sampling are nonprobability sampling techniques that are used to select a sample of individuals from a population (Etikan et al., 2017). According to Etikan et al. (2017), nonprobability sampling is effective when the population is large, and randomization is not possible. This sampling is beneficial when the researcher does not have adequate time or resources.

Samples that can be generalized and fully represent the greater population are costly, time-consuming, and hard to retrieve. A comparison between convenience and representative population-based samples in experiments showed that over 70% of the comparisons yielded the same significance in statistics (Sedgwick, 2013). This shows that convenience sampling does offer the benefit of being cost-effective, easily accessible, and is comparable to samples that are chosen as a representation of specific criteria. However, this research used purposive sampling.

3.8.2 Sample Size and Sample

A sample is the respondents that are chosen from the population for study (Bacon-Shone, 2015). A sample that utilizes data collection techniques will produce results that are valid, reliable, and can be generalized; it will also save resources (Bartlett, Kotrlik, & Higgins, 2001).

Suitable sample size is important for arriving at reliable results and conclusions (Memon, Ting, Cheah, Thurasamy, Chuah & Cham, 2020). According to El-Gilany (2018), a sample size of between 10 and 30 respondents is sufficient for quantitative research. If the sample size is too small, this will question the validity of the results. There is a need for the researcher to secure a sample size that will be able to represent findings that are generalized for individuals who are exposed to agile methodologies.

Memon et al. (2020) stated that any attempt to increase the sample size by just trying to rigorously gain more respondents is less important when compared to the way data is collected. This indicates that trying to increase the sample size without proper consideration for the factors in the selection of the targeted respondents is not important or helpful to the research. The sample size can also be derived from the type of data analysis that will be performed

(Memon et al., 2020). Hair, Black, Babin, & Anderson (2010) proposed that regression analysis requires a minimum of 50 samples, while most research analysis requires 100 samples. Martin & Bateson (1986) indicate that by increasing the sample size, the statistical power is improved up to a level. They also stated the sample size needs to be also weighed up against time and financial constraints. To achieve an acceptable statistical power, a sample size of 200 agile practitioners or those individuals who have had some experience or exposure to agile methodologies will be targeted. Roscoe (1975) stated that there is little backing or justification for a sample size of less than 30 and a sample size of more than 500.

3.9 DATA COLLECTION METHODS

This study was based on a quantitative research approach, which utilized a survey for collecting data. Most interviews are face to face or they use a telephonic survey (Bacon-Shone, 2015). However, the face-to-face interview did not suit this research as the researcher did not want to intimidate the respondent and create biased responses. The questionnaires were used as the survey tool and were distributed to the potential respondents. This questionnaire consisted of closed-ended questions. All of the questions will be based on the Likert scale. The Likert scale which is a 5 or 7-point scale is utilized by respondents to choose the option by rating whether they agree or disagree with statements and was developed by Rensis Likert in 1932 (Sullivan & Artino, 2013). The questionnaire comprised of importance questions and performance questions. The importance questions were designed to gather the perception of the respondent on how they felt about factors that affect agile success. The performance questions were designed similarly, but to find out how these critical success factors were implemented in agile projects. This ties to the research objective to find the link between the importance of the critical factors and the performance of those factors. The questionnaire included options of Strongly Disagree, Disagree, Neutral, Agree, and Strongly Agree for the Importance questions. The questionnaire included options of None, Poor, Fair, Good, and Very Good for the Performance questions. The researcher's objective was to gauge the respondents' views and how the critical factors translated into real-life projects. Google Forms was utilized for capturing the questionnaire and the link was shared with potential participants.

Invitations to complete the questionnaires were bulk emailed to the 192 potential Spar Group Ltd respondents at the SGIS (Spar Group Information Systems) office in Durban with a link to the Google Forms questionnaire. The researcher was not allowed to send the questionnaire individually to understand which individuals were working on agile projects or not. It was difficult to ascertain whether all potential participants would respond timeously. Less than 10 individuals responded from Spar, therefore the researcher opted to use the LinkedIn Professional Network which is a social media platform for professionals. The questionnaire was distributed on the LinkedIn Professional Network to only willing participants. Each LinkedIn member that was contacted was allowed to accept the invitation to participate before the link to the survey was sent. In this way, the researcher was able to send the questionnaire to only those individuals who were willing to participate questionnaire.

The primary data was collected using the quantitative method, however, there are other main methods like interviews, focus group interviews, observation, case studies, and surveys (Kabir, 2016). Linkedin professionals with various titles like Testers, Business Analysts, Project Managers, Cloud specialists, Test Automation personnel, Scrum Masters, Software Developers, and other Information Systems professionals were chosen in line with purposive sampling. The researcher sought professionals who were exposed to agile methodologies that have become popular in development projects. Keywords example, “agile”, “agile methodologies”, “agile software development”, “agile champion”, etc. were used to search for potential respondents on LinkedIn. Each individual was contacted to seek permission before forwarding a link to the questionnaire. The researcher was able to keep track of the respondents by not sending them to those who did not agree to participate. Statisticians believe that a sample size of 100 is the smallest sample size that can be used to achieve meaningful results (Bullen, 2014).

3.10 DATA QUALITY CONTROL

Data quality is data that is relevant to meet the requirements of the user or is appropriate for its purpose (Alizamini, Pedram, Alishahi & Badie, 2010). Quality is generally explained in terms of validity which is the degree to which a research instrument measures what it intends or proposes to measure, while reliability is the degree to which a research instrument can produce the same results if the process is repeated (Taber, 2013). The quality control procedure for item nonresponse is indicative of item nonresponse error which is a function of the item nonresponse rate and the number of respondents and nonrespondents. Quality assurance procedures arise from three types of unit nonresponse which are non-contacts, refusals, and the individual’s inability to partake (Liu, 2012). Non-contacts are the inability to contact the individual, refusals are the individual's decline of the survey participation and the individual's inability to partake could be due to illness, language constraints, not being available, etc. (Biemer & Lyberg, 2003). The researcher contacted each targeted individual to ascertain their interest in this research study. This strengthened the data quality as the link to the questionnaire was only sent to those individuals who asked for it. Those who were uninterested, busy, or were unable to participate did not respond but some did respond to elaborate why they could not participate due to work schedules. The questionnaire was completely anonymous which allowed the respondents to feel free to answer without bias. Some respondents inquired whether the questionnaire would have any impact on their jobs. They were reassured that it was an anonymous questionnaire, and the researcher would not be able to link the responses with any individual or organisation as personal details were not requested. This also increased the data quality as the respondents were at ease to complete the questionnaire. The questionnaire was not designed to collect personal details or confidential information about the individual or their organisations.

The questions were also compared to another questionnaire that was used in a previous study by Xu (2003) to validate that the questions were adhering to the required measures. The design of the questionnaire was vetted against this sample questionnaire to ensure that the importance and performance of CSFs could be accurately measured as the sample had importance and performance questions. A pilot study was then conducted with a small group of participants who were knowledgeable in agile and also worked in an agile development environment (Doody & Doody, 2015). Pilot studies allow researchers to investigate the effectiveness of their intended data collection and analysis methods and to gain practice for the actual study (Doody & Doody, 2015). The researcher emailed the questionnaire to two agile practitioners to confirm that the questionnaire was readable, logical, and applicable to agile projects. They were asked to confirm that the questions were clear, logical, and applicable to agile projects. The questions were confirmed to be reliable (clear) and valid (applicable to agile projects). This strengthened the validity of the questionnaire.

The questionnaire was captured on Google Forms, an online survey tool which also strengthened the validity as the researcher did not have to capture the data again to analyze using SPSS. This allowed for no chance of errors in capturing the data for analysis. The data collected in the Google Forms survey was exported to Microsoft Excel and imported directly into SPSS (no manual capture) and analyzed. The lack of manual capture required strengthened the validity of the data.

Cronbach's alpha is a statistical value that is used to indicate whether the data collected and tests conducted in research studies are reliable, fit the purpose, and can be trusted (Taber, 2018). It is the most common measure of reliability or internal consistency which is generally used with Likert scale questions in a questionnaire and if the reliability of the scale is to be determined (Cronbach's Alpha in SPSS Statistics - procedure, output and interpretation of the output using a relevant example | Laerd Statistics., 2018). Cronbach's alpha was therefore computed in this research to determine the reliability of the questionnaires.

3.11 ETHICAL CONSIDERATIONS

The ethical considerations that will be covered will enable the respondents to feel free and not forced into completing the survey. The rights of the individual will not be infringed, and the respondents sign an acknowledgment of consent to complete the survey to ensure that they were not coerced into the activity. Since the questionnaire was created on Google Forms, the respondent was able to select whether they wanted to participate or not. The questionnaire was completely anonymous; therefore the respondent was not required to enter their details example, Name, Surname, or Organisations. This fulfilled the ethical requirement that the researcher had in mind to ensure that the respondents were not apprehensive to complete the questionnaire due to loyalty to their organisations. The intention was to create an honest survey whereby the respondents did not feel that their jobs or reputation could be affected by their responses. This allowed the respondents to answer freely and honestly as the researcher did not have their identity attached to the responses. Further, this strengthened the validity of the results as the respondents were not under duress when answering the questionnaire. This acknowledgment of consent will serve as a formal indication that will highlight the purpose and intention of the research and serve as proof that the participants were not coerced into completing the surveys and it was of their own accord. The identity of the respondents will remain confidential. The respondents were not forced or harassed into participation. It was a research study that will strongly uphold the privacy and rights of the participants. The researcher ensured that the respondents understand the reason for the survey, the objectives of the study and that their participation is purely at their own will and discretion for information gathering. The questionnaire was accessed via the Google Forms weblink and the summary section explained the details of the research, the topic, the ethical clearance obtained, and the informed consent were included. A disclaimer was added to state that the respondent was not required to enter or sign the informed consent online as the survey was anonymous. However, the informed consent was fulfilled by the questionnaire allowing the potential respondent to choose whether they wanted to participate in the research or not by a selection option on the questionnaire.

Ethical clearance was obtained from the UKZN Humanities and Social Research Ethics Committee. A gatekeeper letter was obtained from Spar Group Ltd to be able to distribute the questionnaires to the professionals in their SGIS department. Unfortunately, the response rate was very low from Spar with less than 10% of responses and the researcher opted to use LinkedIn professional service to an optimum to obtain as many respondents as possible for a viable research study. To maintain an ethical research environment, each targeted LinkedIn respondent was contacted via LinkedIn messaging to enquire whether they would be interested in participating in the research. Only those individuals who opted to participate were sent the link to the questionnaire. In this way, the researcher was not infringing on the social media rights of the individual, by sending unwanted links without prior permission.

3.12 DATA ANALYSIS

Statistics involves creating and learning various procedures for gathering, representing data, and analyzing data and this field is made up of two areas which are descriptive and inferential statistics (Singh, 2018). Descriptive statistics entails describing the main characteristics or attributes of the data using the dispersion measures like standard deviation, variance, and range, and the measures like median, mean, or mode (Singh, 2018). This data can be represented using charts, tables, or graphs. Inferential statistics entails using sample data and drawing conclusions from this data on the greater public (Singh, 2018).

This study used both descriptive and inferential statistics. Descriptive statistics were used via SPSS to analyze the frequency tables. Inferential statistics were used in the form of the t-test. Common methodologies that can be used are Anova, Chi-square, t-test, etc. The Anova test is used to determine which factors affect the respondents more than others (ANOVA - Statistical Test - The Analysis Of Variance, 2000). The Chi-square test can compare data with an underlying difference, for example, boys and girls in learning environments as they are inherently different (Department of Linguistics - Home | Department of Linguistics, 2008) The

Anova test and Chi-square test were not suitable for this research. The Anova test was not selected as this test is suitable for comparing more than two means. The Chi-square test was not suitable for this research as there was no underlying difference between the respondents as the perceived importance and perceived performance applied to the same sample of respondents. The t-test is an inferential statistic test that is utilized to find out if the averages of the two groups have a considerable difference. The decision was made to use a t-test as the preferred test for the data analysis. This test was used to ascertain whether there was a significant difference between the perceived importance and perceived performance of the critical success factors. Inferential statistics aims to arrive at conclusions of a large population from a sample set and, in this study, the data collected will be analyzed by inferential and descriptive statistics. Inferential statistics will enable the researcher to arrive at conclusions about the data concerning the greater population of agile developers in South Africa, while descriptive statistics will allow the researcher to describe the data in a meaningful way.

By calculating and analyzing descriptive statistics, it is possible to view basic information about the population sample and to look for abnormalities in the data that has been collected. The descriptive statistics analysis via SPSS frequency tables, the paired sample t-test, the relative importance index (RII), and Cronbach's test were used for this research and the results were reported. SPSS v.27 was used to calculate the frequency tables for the Importance and Performance CSFs and Microsoft Excel was used for the subsequent calculations.

The RII was calculated as follows:

The data collection method of questionnaires led to the analysis using the Relative Importance Index which is appropriate for a Likert-scale questionnaire.

The frequency for each question is multiplied by the weighting for that response. Strongly Agree has a weighting of 5, Agree (4), Neutral (3), Disagree (2), and Strongly Disagree (1). The weighting is then added for each factor, then divided by the total number of responses multiplied by the highest weighting which is 5. This computes the RII % and the ranking can be established.

The formula for RII:

$$RII = \frac{\Sigma W}{(A*N)} = \frac{5n5 + 4n4 + 3n3 + 2n2 + 1n1}{(A*N)}$$

RII = Relative Importance Index; A = highest weight which is 5 in the case of the Likert scale scores and N = total number of respondents.

Another study by Hatkar & Hedaoo (2016), used the RII index to determine the ranking of factors that contributed to delays in construction. Regression coefficients are not sufficient as agile practitioners may not consider the incremental amount of importance that one critical success factor has, while the other CSFs are kept constant (Hossen, Kang & Kim, 2015). They would instead consider all the factors that are important at once and give a weighting to each factor relative to the rest to determine the overall importance (Hossen et al., 2015). The RII is a statistical method that is used to ascertain the ranking or order of importance of various factors (Hossen et al., 2015). This was the rationale behind the choice of RII for this research as the ranking of the importance and performance of the CSFs covered two objectives of the research.

The survey questionnaire was developed to incorporate the analysis by RII as the Likert scale questions were suitable to an analysis by RII. The items on the survey were categorized into the constructs of importance and performance and both aspects were designed to be measured on the 5-point Likert scale and analyzed using RII. The design allowed for the importance questions to be answered in terms of how much on the Likert scale did the respondent agree with the statements and then how well the factors were performed in projects by the respondents rating of them.

The data was analyzed using this RII method to calculate the RII index, to rank the CSFs based on the RII, and ascertain the correlation on the ranking of the CSFs between the Importance and Performance factors (Hatkar & Hedaoo, 2016). In a study by Gündüz, Nielsen, & Mustafa Özdemir (2013), the Relative Importance Index formula was used. The RIIs were calculated for each factor in this study and the same method was adopted whereby the RIIs were averaged to attain the RII for the particular CSF.

3.13 CONCLUSION

The research methods used in the research study were discussed in this chapter. The research methodology implemented was explained and the reasons for this choice. The closed-ended questionnaires were used as the research tool as quantitative research was chosen as the research method. The research questions were also discussed. The ethical considerations of the study, informed consent, and anonymity were also explained. The presentation of the results will be covered in Chapter 4.

CHAPTER 4

RESULTS

4.1 INTRODUCTION

This chapter outlines the results that were obtained from the data collected using the closed-ended questionnaires that were distributed to agile stakeholders. Statistical Package for Social Sciences (SPSS) v27.0 was used for the descriptive and inferential data analysis and the subsequent statistical tests. A sample size of 200 was targeted and only 110 responses were received which resulted in a response rate of 55%. According to Bullen (2014), statisticians believe that a sample size of 100 is the smallest sample size that can be used to achieve meaningful results.

The demographic data were analyzed using descriptive analysis, the importance factors were analyzed using descriptive analysis and then the critical factors were analyzed using the Relative Importance Index (RII). The performance of the factors in the organisation was analyzed using descriptive analysis and thereafter the gap between the importance and performance of the critical success factors was evaluated using the Paired Samples T-Test.

4.2 DESCRIPTIVE STATISTICS

4.2.1 Demographic factors of respondents

Table 4.1 Demographical statistics of respondents

Gender					
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Female	35	31,8	31,8	31,8
	Male	75	68,2	68,2	100,0
	Total	110	100,0	100,0	
Age Group					
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	21-25	2	1,8	1,8	1,8

	26-30	17	15,5	15,5	17,3
	31-35	22	20,0	20,0	37,3
	36-40	18	16,4	16,4	53,6
	41-45	23	20,9	20,9	74,5
	Above 45	28	25,5	25,5	100,0
	Total	110	100,0	100,0	
Level of tertiary education					
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Certificate	8	7,3	7,3	7,3
	Degree	67	60,9	60,9	68,2
	Diploma	26	23,6	23,6	91,8
	None	1	0,9	0,9	92,7
	Other	8	7,3	7,3	100,0
	Total	110	100,0	100,0	

Table 4.1 shows the demographic information of the respondents for the Gender, Age, and Tertiary Education factors, respectively. A sample of 200 respondents was targeted, but 110 responses were received. This resulted in seventy males (68%) and 35 females (32%) who participated in this study. Twenty-six percent of the respondents were 45% years and above, 21% were between 41-45 years, 20% were between 31-35 years, 7% were between 36-40 years, 16% were between 26-30 years, and 1.8% between 21-25years. Sixty-one percent of the respondents attained a degree qualification, 23.6% obtained diplomas, 7.3% obtained certificate qualifications, 7.3% indicated “Other” as a qualification but did not elaborate, and only 0.9% represented 1 respondent with no qualification.

4.2.2 Type of agile environment

Table 4.2 Demographic statistics of Agile development, Location, Number of project team members, Number of projects

Type of agile development				
	Frequency	Percent	Valid Percent	Cumulative Percent

Valid	Adaptive Software Development	2	1,8	1,8	1,8
	Dynamic Systems Development Method	3	2,7	2,7	4,5
	Extreme Programming	1	0,9	0,9	5,5
	Feature-Driven Development	9	8,2	8,2	13,6
	Lean software development	5	4,5	4,5	18,2
	Scrum	90	81,8	81,8	100,0
	Total	110	100,0	100,0	
Organisation location					
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Free State	1	0,9	0,9	0,9
	Gauteng	50	45,5	45,5	46,4
	KwaZulu Natal	40	36,4	36,4	82,7
	Western Cape	19	17,3	17,3	100,0
	Total	110	100,0	100,0	
Number of team members in projects					
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	10-15 team members	21	19,1	19,1	19,1
	5 or less team members	12	10,9	10,9	30,0
	5-10 team members	39	35,5	35,5	65,5
	More than 15 team members	38	34,5	34,5	100,0
	Total	110	100,0	100,0	
Number of agile projects that you were involved in					
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	0-5	27	24,5	24,5	24,5
	11-14	14	12,7	12,7	37,3
	15-20	13	11,8	11,8	49,1

	21-25	2	1,8	1,8	50,9
	6-10	23	20,9	20,9	71,8
	Above 25	31	28,2	28,2	100,0
	Total	110	100,0	100,0	

Table 4.2 shows that the majority of the respondents selected Scrum (81.8%) as the type of agile environment that they work in, Feature-Driven Development accounting for 8.2% of the sample, Lean software development (4.5%), Dynamic Systems Development Method (2.7%), and Adaptive Software Development with 1.8% of the sample. It is interesting to note that Extreme Programming accounts for only 0.9% of the sample with only 1 response. In a study by Chiyangwa (2017), XP and Scrum were noted as being the most popular from the sample responses. Scrum is by far the most popular in this study.

4.2.3 Organisation Location

Table 4.2 shows that most of the respondents are members of organisations in Gauteng (45.5%), followed by 36.4% in KwaZulu Natal, 17.3% in Western Cape, 0.9 % in Free State, and 0 respondents in each of Mpumalanga, Eastern Cape, Northern Cape, and Limpopo.

4.2.4 Number of members in the agile project team

Table 4.2 shows that most respondents (35.5%) reported that they were in teams of 5-10 members, followed by 34.5% with more than 15 members, 19.1% with 10-15 team members, and 10.9% with 5 or fewer members. 53.6% of respondents are in teams of 10 or more members.

4.2.5 Number of agile projects (agile maturity)

Table 4.2 shows that the majority of respondents (28.2%) reported that they were involved in more than 25 projects, followed by 24.5% with 5 or fewer projects, 20.9% with 6-10 projects, 12.7% with 11-14 projects, 11.8% with 15-20 projects and 1.8% with 21-25 projects.

4.2.6 Organisation

Table 4.3 Demographical statistics of Organisation, Role, and years of experience

Industry Type of Organisation					
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Banking	16	14,5	14,5	14,5
	Consulting services	13	11,8	11,8	26,4
	Distribution	2	1,8	1,8	28,2
	Financial/accounting	1	0,9	0,9	29,1
	Health Care	1	0,9	0,9	30,0
	Information Services	42	38,2	38,2	68,2
	Information technology	2	1,8	1,8	70,0
	Insurance	1	0,9	0,9	70,9
	Lead Generation & Marketing	1	0,9	0,9	71,8
	Legal	2	1,8	1,8	73,6
	Manufacturing	1	0,9	0,9	74,5
	Mining	1	0,9	0,9	75,5
	Municipal/Government	2	1,8	1,8	77,3
	Online Gaming	7	6,4	6,4	83,6
	Online Software	1	0,9	0,9	84,5
	Retail	11	10,0	10,0	94,5
	Telecommunications	5	4,5	4,5	99,1
	Various	1	0,9	0,9	100,0
	Total	110	100,0	100,0	

Your role in the organisation/project					
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Business Analyst	3	2,7	2,7	2,7
	Client	1	0,9	0,9	3,6
	Data Analyst	2	1,8	1,8	5,5
	Developer/Tester	30	27,3	27,3	32,7
	Functional Manager	6	5,5	5,5	38,2
	Organisation management	8	7,3	7,3	45,5
	Other (Please specify)	15	13,6	13,6	59,1

	Project Manager	10	9,1	9,1	68,2
	Scrum Master	29	26,4	26,4	94,5
	Team Lead	6	5,5	5,5	100,0
	Total	110	100,0	100,0	

Number of years' experience in agile software development projects					
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	0-2	12	10,9	10,9	10,9
	12-14	9	8,2	8,2	19,1
	3-5	25	22,7	22,7	41,8
	6-8	30	27,3	27,3	69,1
	9-11	16	14,5	14,5	83,6
	Above 14	18	16,4	16,4	100,0
	Total	110	100,0	100,0	

Table 4.3 shows that 39.1% of the organisations are Information Services, followed by Banking (14.5%), Consulting Services (11.8%), Retail (10%), and Gaming and Entertainment (6.4%). There was a 0% representation for the Advertising, Hospitality, NGO, and Building/Construction industries.

4.2.7 Respondent Role & experience

Table 4.3 shows that the majority of the respondents have the roles of Developer/Tester (27%) and Scrum Master (26%) which accounts for 53% of the sample. This is followed by Other which accounts for 14% of the sample, 9% which represents Project Manager, 7% which represents Organisation management, 5% which entails Functional Managers and Team Lead each, 3% were Business Analysts, 2% were Data Analysts, 1% were clients and the respondents did not specify the context of Other.

4.2.8 Agile project experience

Table 4.3 shows that the majority of respondents (27.3%) had 6-8 years' experience in agile projects, followed by 22.7% with 3-5 years' experience, 16.4% with over 14 years' experience, 14.5% with 9-11 years' experience, 8.2% with 12-14 years' experience and 10.9% with 0-2 years' experience.

4.3 CRITICAL SUCCESS FACTORS

4.3.1 Descriptive analysis of importance factors

4.3.1.1 Organisational Structure

According to Abdalhamid & Mishra (2017), when an organisation adopts agile methodologies, the organisation has to restructure to ensure the appropriate shift in its culture and way of working. The following figure shows the results obtained for each of the ten statements that were presented on the importance factor of organisational structure.

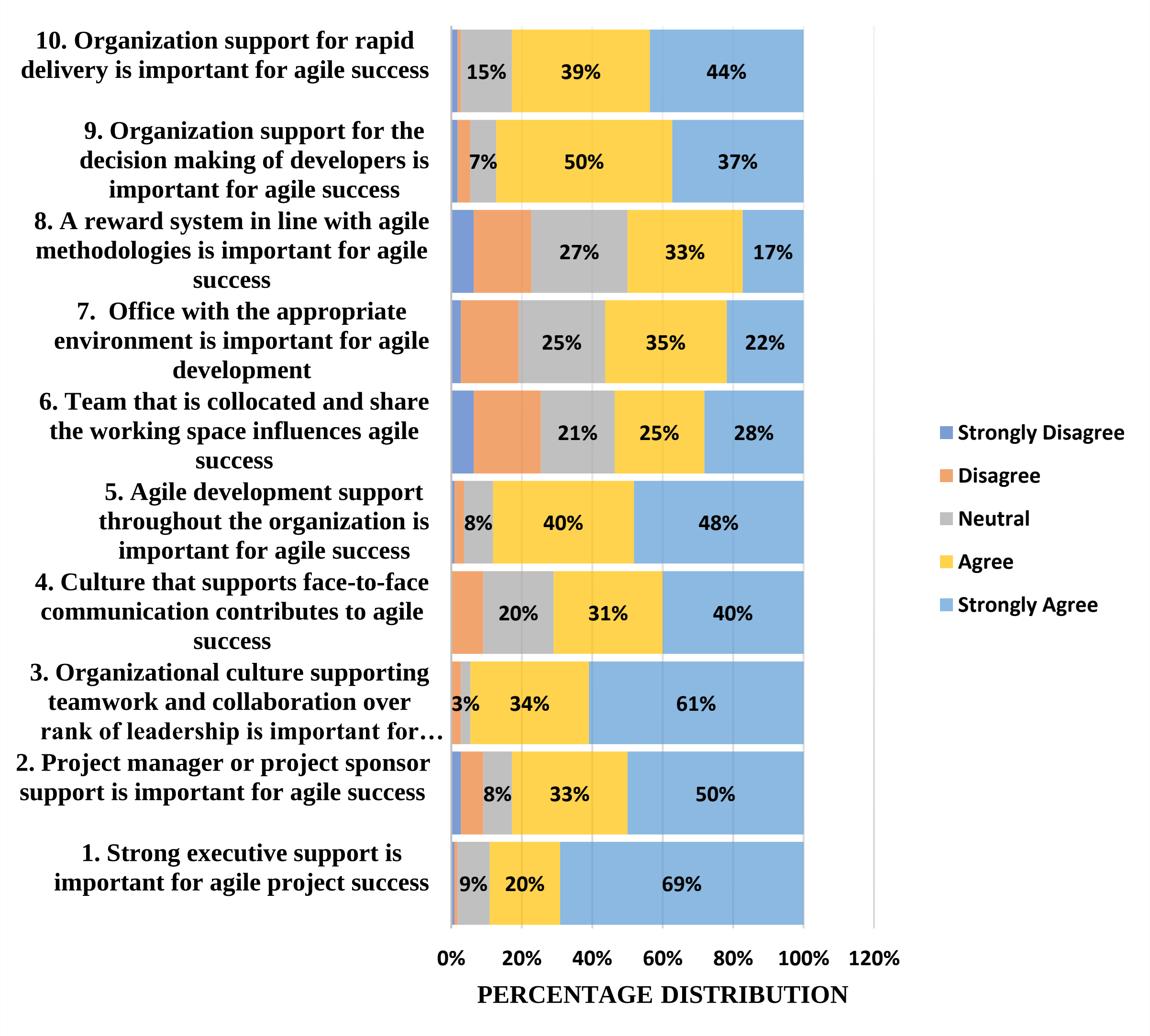

Figure 4.1 Descriptive Statistics for Organisational Structure

Figure 4.1 indicates that 95% agreed that organisational culture supporting teamwork and collaboration over the rank of leadership is important for agile success, 89% (strongly agree and agree) of the respondents agreed that strong executive support is important for agile success, 88% of respondents agreed that agile development support throughout the organisation is important for agile success, 87% agreed that organisation support for the decision making of developers is important for agile success, 83% agreed that project manager or project sponsor support is important for agile success, and 83% agreed that organisation support for rapid delivery is important for agile success. Although all the items for this factor were considered important for agile success, the items described above show the highest agreement scores. Therefore, this signifies that the majority of the agile practitioners agreed that the organisational factor is important for agile success. This is further substantiated by a mean value of 4.05 for this factor.

4.3.1.2 People Factor

According to Chow and Cao (2008), people factors include the capabilities of the team members and the relationship with the customer. The people dimension indicates the following results.

The following figure shows the results obtained for each of the seven statements that were presented on the importance factor of people. The purpose was to analyze the responses received which indicated the importance of the people factor for agile project success as perceived by the respondents.

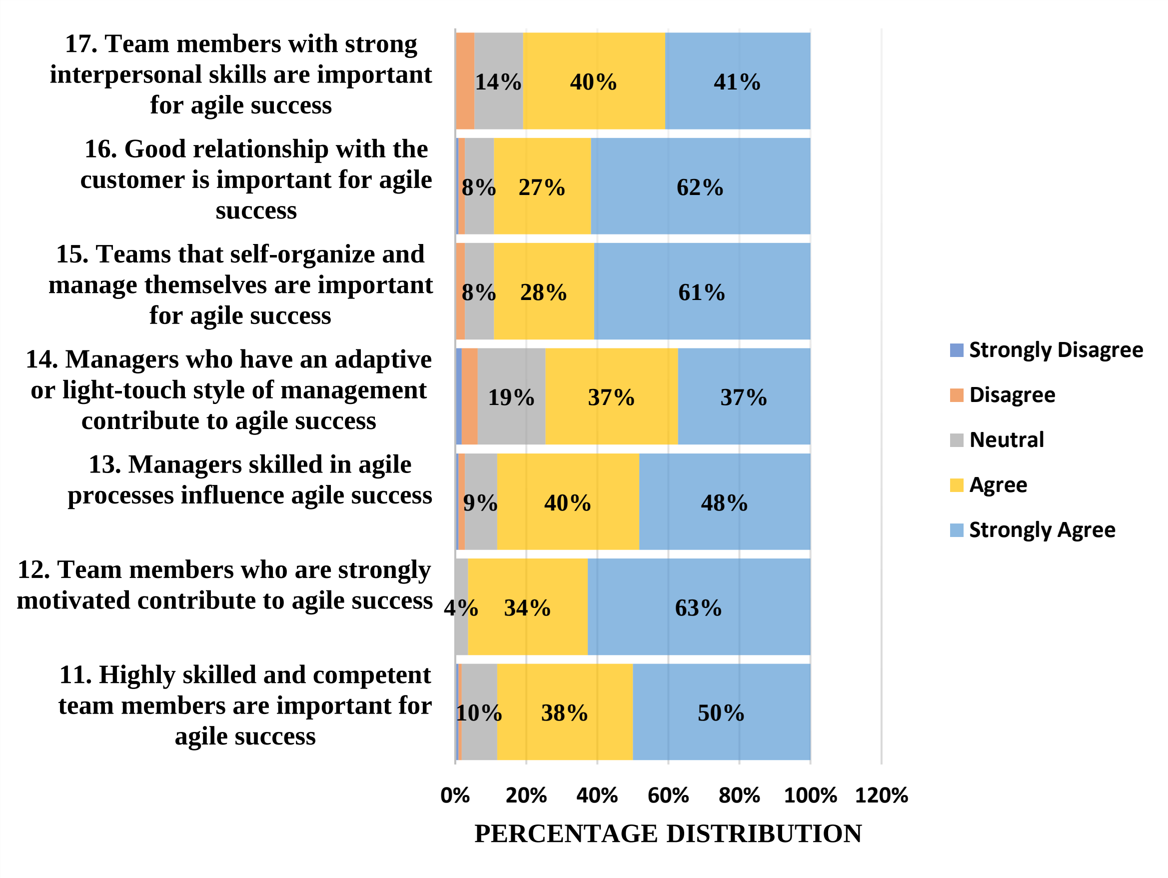

Figure 4.2 Descriptive Statistics for People

Figure 4.2 indicates that 97% (strongly agree and agree) of the respondents agreed that Team members who are strongly motivated contribute to agile success, 89% agreed that Teams that self-organize and manage themselves are important for agile success, 89% agreed that a good relationship with the customer is important for agile success, 88% agreed that highly skilled and competent team members are important for agile success, 88% agreed that managers skilled in agile processes influence agile success, 81% of respondents agreed that team members with strong interpersonal skills are important for agile success and 74 % agreed that managers who have an adaptive or light-touch style of management contribute to agile success. Each factor was considered to be important for agile success and this implies that agile practitioners view the people factor as important for agile success. The mean of 4.35 further emphasizes the importance.

4.3.1.3 Process factor

Process factors incorporate the process of project management and the process of the project definition (Chow & Cao, 2008). Various agile methodologies exist and it is imperative to select the appropriate method for a project and ensure that it can be integrated with processes that are external (Abdalhamid & Mishra, 2017).

The following figure shows the results obtained for each of the six statements that were presented on the importance factor of process. The purpose was to analyze the responses received which indicated the importance of the process factor for agile project success as perceived by the respondents.

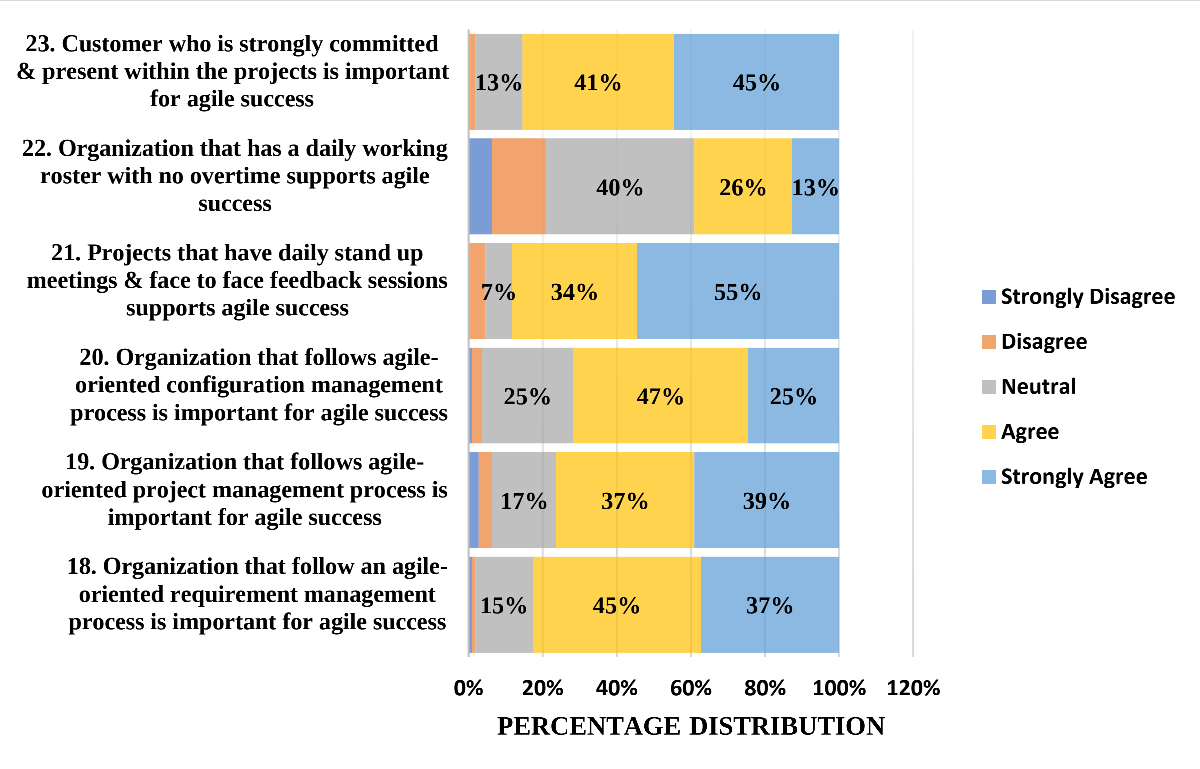

Figure 4.3 Descriptive Statistics for Process

Figure 4.3 indicates that 89% (strongly agree and agree) of the respondents agreed that projects that have daily stand-up meetings & face to face feedback sessions support agile success, 86% agreed that the customer who is strongly committed and present within the projects is important for agile success, 76% agreed that organisations that follow agile-oriented project management processes are important for agile success and 72% agreed that organisations that follow an agile-oriented configuration management process are important for agile success. Each factor was considered to be important for agile success and this implies that agile practitioners view the process factor as important for agile success. The mean of 4.01 further emphasizes that most of the respondents indicated that this factor was important to agile success.

4.3.1.4 Technical Factors

According to Chow and Cao (2008), technical factors include defining coding factors at the outset of the project, delivering the most pertinent attributes first, an integration that is continuous and correct, proper training rendered to the team in technical aspects, and delivery of software in regular intervals.

The following figure shows the results obtained for each of the nine statements that were presented on the importance factor of the technical factor. The purpose was to analyze the responses received which indicated the importance of the technical factor for agile project success as perceived by the respondents.

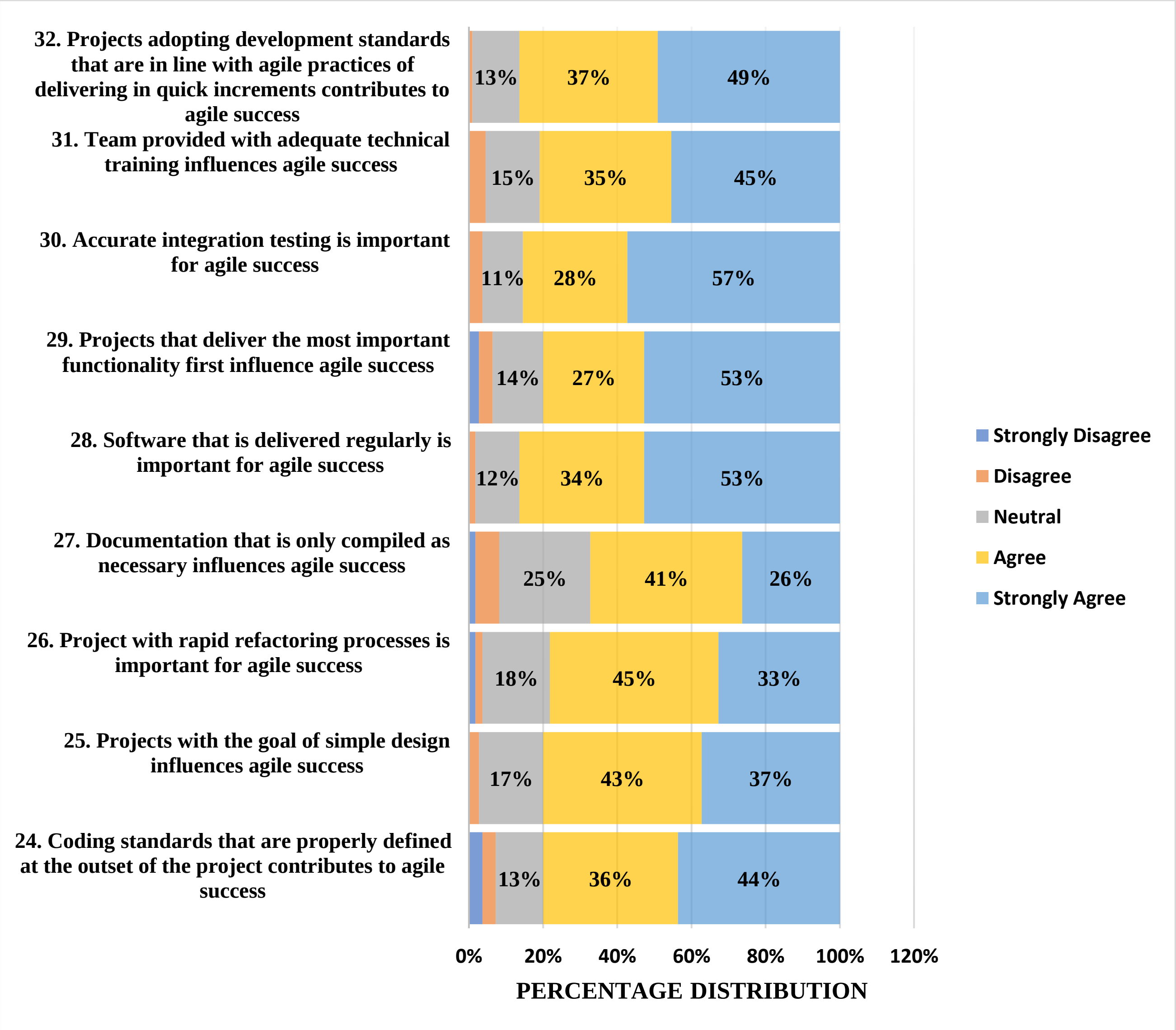

Figure 4.4 Descriptive Statistics for Technical Factors

Figure 4.4 indicates that 87% (strongly agree and agree) of respondents agreed that software that is delivered regularly is important for agile success, 86% agreed that projects adopting development standards that are in line with agile practices of delivering in quick increments contribute to agile success, 85% agreed that accurate integration testing is important for agile success, 80% agreed that predefined coding standards influence agile success, 80% also agreed that simple design leads to agile success, 80% agreed that delivering important functionality first influence agile success, while 80% also agreed that teams provided with adequate technical training influences agile success. Only 67% believed that documentation that is only compiled as necessary is important for agile success. The majority of technical factors were considered to be important for agile success and this implies that agile practitioners view the technical factor as important for agile success. The mean of 4.20 further emphasizes the importance of the technical factor.

4.3.1.5 Project

Project factors include the nature of the project, the type of project, and the schedule (Chow & Cao, 2008).

The following figure the results obtained for each of the seven statements that were presented on the importance factor of project. The purpose was to analyze the responses received which will indicate the importance of the project factor in agile project success as perceived by the respondents.

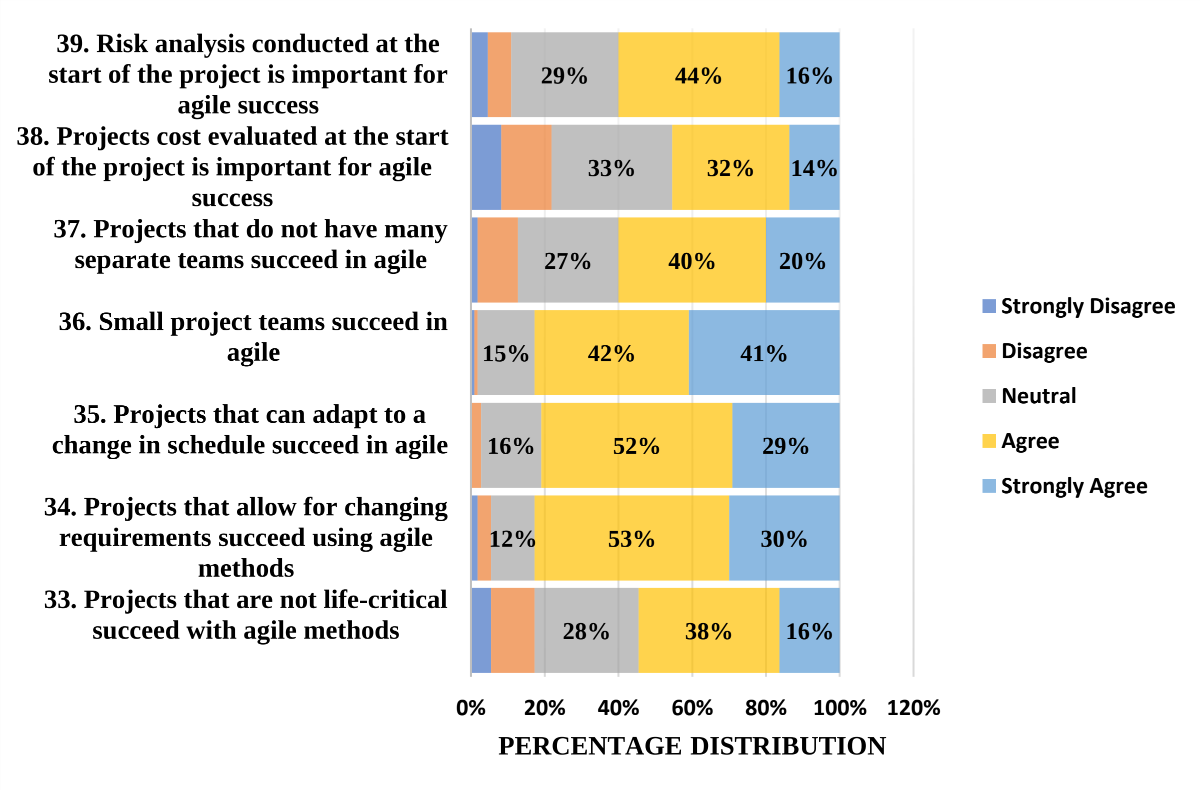

Figure 4.5 Descriptive Statistics for Project

Figure 4.5 indicates that 83% (strongly agree and agree) of the respondents agreed that projects that allow for changing requirements succeed using agile methods, 83% also agreed that small project teams succeed in agile, 81% agreed that projects that can adapt to a change in schedule succeed in agile, 45% agreed that if the project cost is evaluated at the start of the project, it is important for agile success. The mean of 3.77 indicates that some factors were considered important while the other factors were not as important which resulted in a mean below 4.

4.3.1.6 Test Automation

Test Automation enables manual testing to be automated by utilizing automation tools and the automated tests are quicker and can be repeated (Mahajan, Shedge & Patkar, 2019). Further, it reduces costs as the tests can be performed quicker than by humans (Mahajan et al., 2019).

The following figure shows the results obtained for each of the five statements that were presented on test automation as an importance factor in agile project success. The purpose was to analyze the responses received which indicated the importance of test automation as perceived by the respondents.

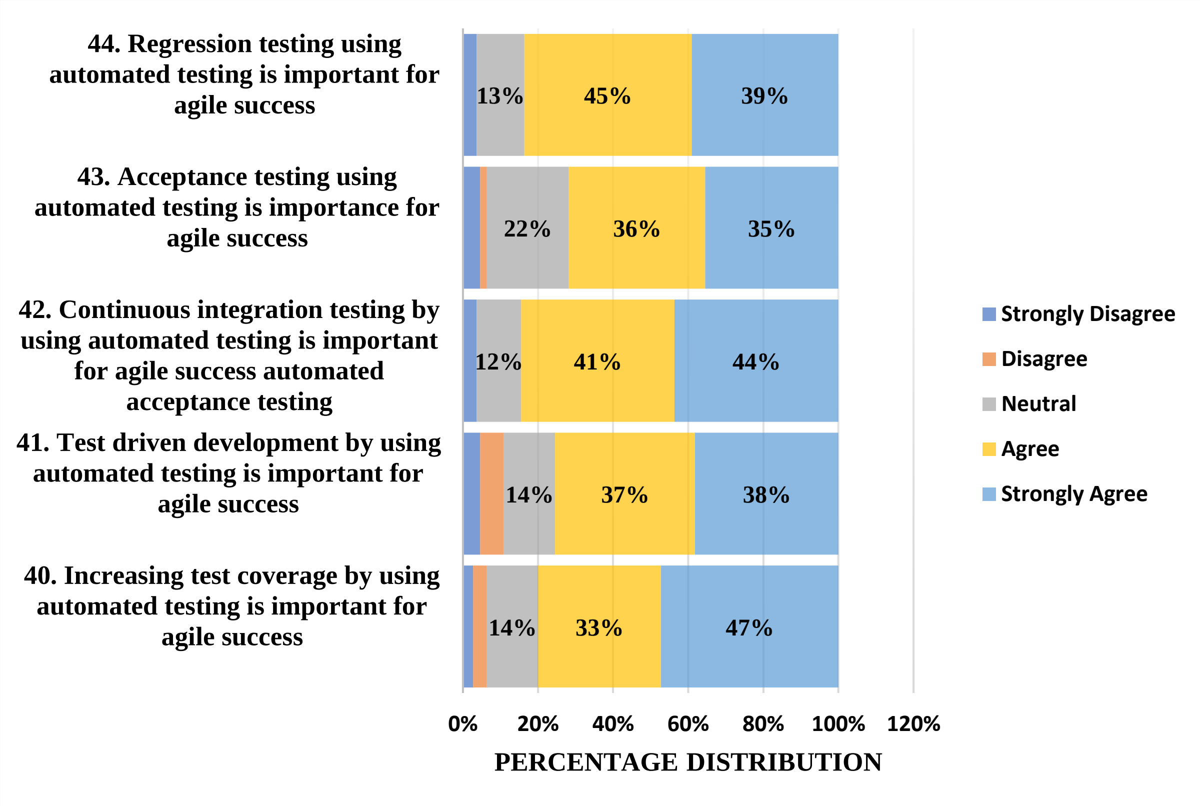

Figure 4.6 Descriptive Statistics for Test Automation

Figure 4.6 indicates that 84% (strongly agree and agree) of respondents agree that regression testing using automated testing is important for agile success, 85% agreed that continuous integration testing by using automated testing is important for agile success and 80% agreed that increasing test coverage by using automated testing is important for agile success. The majority of the factors were viewed as important which indicates that test automation is important for agile success. The mean of 4.10 further emphasizes the importance of test automation for agile success.

4.3.1.7 Cloud Computing

Cloud computing is a completely web-based concept and is made up of a computer network that services architecture to access software and data (Sayeed et al., 2017). Customers can have access to software applications with cloud computing via any area that has internet available (Kalem et al., 2013). The developing aspects of cloud computing in agile teams are used to accelerate the time in which software is developed (Sayeed et al., 2017).

The following figure shows the results obtained for each of the five statements that were presented on the importance factor of cloud computing in the success of agile projects. The purpose was to analyze the responses received which indicated the importance of the cloud computing factor as perceived by the respondents.

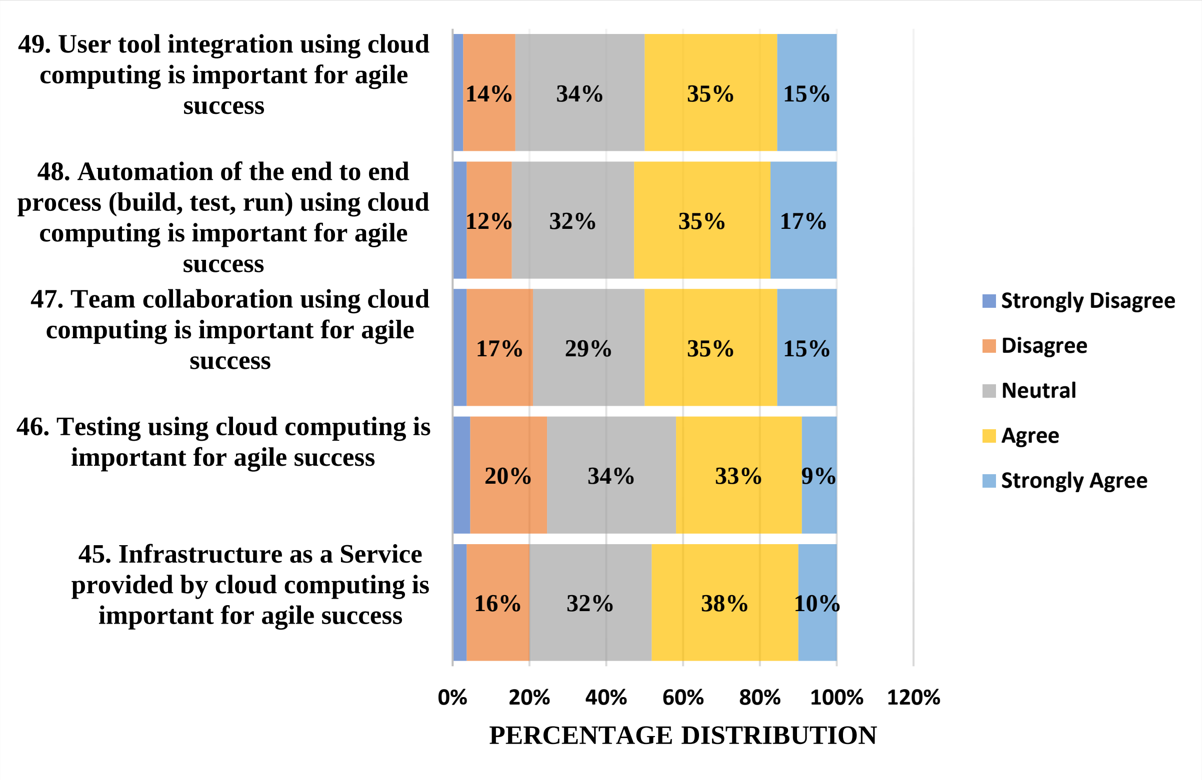

Figure 4.7 Descriptive Statistics for Cloud Computing

Figure 4.7 indicates that 52% (strongly agree and agree) of the respondents indicated that automation of the end to end process (build, test, run) using cloud computing was important for agile success, 50% agreed that team collaboration using cloud computing was important for agile success and 50% agreed that user tool integration using cloud computing was important for agile success. Approximately 50% of the respondents indicated that cloud computing is important for agile success which does not account for the majority. The mean of 3.39 further emphasizes that the majority of respondents do not view cloud computing as important for agile success.

The descriptive analysis indicated that certain factors are perceived by agile stakeholders to be more important to agile success than others. Further, the items within each factor also showed varying degrees of agreement. The next section shows the ranking of these factors and items using the relative importance index.

4.3.2 Critical success factors that are perceived to be most important by stakeholders for agile success.

4.3.2.1 Organisation

Table 4.4 shows the RII and ranking of the ten statements from the questionnaire on the Organisation CSFs

Table 4.4 RII and Ranking of Organisational Structure CSFs

Organisational structure	**RII**	**Rank**
1. Strong executive support is important for agile project success	**0,911**	**1**
3. Organisational culture supporting teamwork and collaboration over the rank of leadership is important for agile success	**0,905**	**2**
5. Agile development support throughout the organisation is important for agile success	**0,864**	**3**
10. Organisation support for rapid delivery is important for agile success	0,844	4
2. Project manager or project sponsor support is important for agile success	0,842	5
9. Organisation support for the decision making of developers is important for agile success	0,835	6
4. Culture that supports face-to-face communication contributes to agile success	0,804	7
7. Office with the appropriate environment is important for agile development	0,713	8
6. Team that is collocated and share the working space influences agile success	0,700	9
8. A reward system in line with agile methodologies is important for agile success	0,676	10

. The top 5 items within the organisational factor are Strong executive support (RII = 0.911), Organisational culture supporting teamwork and collaboration over the rank of leadership (RII = 0.905), Agile development support throughout the organisation (RII = 0.863), Organisation support for rapid delivery (RII = 0,843) and a Project manager or project sponsor support (RII = 0,841).

4.3.2.2 People

Table 4.5 shows the RII and ranking of the seven statements from the questionnaire on the People CSFs

Table 4.5 RII and Ranking of People CSFs

People	RII	Rank
17. Team members with strong interpersonal skills are important for agile success	**0,745**	**1**
15. Teams that self-organize and manage themselves are important for agile success	**0,744**	**2**
16. Good relationship with the customer is important for agile success	**0,740**	**3**
14. Managers who have an adaptive or light-touch style of management contribute to agile success	0,733	4
13. Managers skilled in agile processes influence the agile success	0,687	5
11. Highly skilled and competent team members are important for agile success	0,691	6
12. Team members who are strongly motivated contribute to agile success	0,729	7

The most influential items for the people factor were Team members who are strongly motivated contribute to the agile success (RII = 0.918), Teams that self-organize and manage themselves are important for agile success (RI = 0.895), Good relationship with the customer is important for agile success (RII = 0.895), Highly skilled and competent team members are important for agile success (RII = 0.871), Managers skilled in agile processes influence the agile success (RII = 0.865) and Team members with strong interpersonal skills are important for agile success (RII = 0.833).

4.3.2.3 Process

Table 4.6 shows the RII and ranking of the six statements from the questionnaire on the Process CSFs

Table 4.6 RII and Ranking of Process CSFs

Process	RII	Rank
21. Projects that have daily stand up meetings & face to face feedback sessions supports the agile success	**0,838**	**1**
19. Organisation that follows an agile-oriented project management process is important for agile success	**0,707**	**2**
23. Customer who is strongly committed & present within the projects is important for agile success	**0,696**	**3**
18. Organisation that follows an agile-oriented requirement management process is important for agile success	0,693	4
22. Organisation that has a daily working roster with no overtime supports the agile success	0,667	5
20. Organisation that follows agile-oriented configuration management process is important for agile success	0,664	6

The most influential Process factors are Projects that have daily stand up meetings & face to face feedback sessions supports agile success (RII = 0.876), Customer who is strongly committed & present within the projects is important for agile success (RI = 0.856) and Organisations that follow an agile-oriented requirement management process is important for agile success (RI = 0.835). Organisations that have a daily working roster with no overtime support agile success (RI = 0.649) is the least important Process factor.

4.3.2.4 Technical Factors

Table 4.7 shows the RII and ranking of the nine statements from the questionnaire on the Technical CSFs

Table 4.7 RII and Ranking of Technical Factor CSFs

Technical Factors	RII	Rank
30. Accurate integration testing is important for agile success	**0,878**	**1**
28. Software that is delivered regularly is important for agile success	**0,875**	**2**

32. Projects adopting development standards that are in line with agile practices of delivering in quick increments contributes to agile success	**0,869**	**3**
29. Projects that deliver the most important functionality first influence agile success	0,847	4
31. Team provided with adequate technical training influences agile success	0,844	5
25. Projects with the goal of simple design influences agile success	0,829	6
24. Coding standards that are properly defined at the outset of the project contributes to agile success	0,825	7
26. Project with rapid refactoring processes is important for agile success	0,811	8
27. Documentation that is only compiled as necessary influences agile success	0,767	9

The most influential factors are Accurate integration testing is important for agile success (RII = 0.878), Software that is delivered regularly is important for agile success (RII = 0.875), Projects adopting development standards that are in line with agile practices of delivering in quick increments contributes to agile success (RII = 0.869), Projects that deliver the most important functionality first influence the agile success (RII = 0.847) and Team provided with adequate technical training influences agile success (RII = 0.844). The least important factor was Documentation that is only compiled as necessary influences agile success (RII = 0.767).

4.3.2.5 Project

Table 4.8 shows the RII and ranking of the seven statements from the questionnaire on the Project CSFs

Table 4.8 RII and Ranking of Project CSFs

Project	**RII**	**Rank**
36. Small project teams succeed in agile	**0,842**	**1**
35. Projects that can adapt to a change in schedule succeed in agile	**0,815**	**2**

34. Projects that allow for changing requirements succeed using agile methods	**0,811**	**3**
37. Projects that do not have many separate teams succeed in agile	0,731	4
39. Risk analysis conducted at the start of the project is important for agile success	0,722	5
33. Projects that are not life-critical succeed with agile methods	0,696	6
38. Project cost evaluated at the start of the project is important for agile success	0,658	7

The most influential factors are Small project teams that succeed in agile (RII = 0.842), Projects that can adapt to a change in schedule succeed in agile (RII = 0.815), and Projects that allow for changing requirements that succeed using agile methods (0.811). The least important factor was Project cost evaluated at the start of the project is important for agile success (RII = 0.658).

4.3.2.6 Test Automation

Table 4.9 shows the RII and ranking of the five statements from the questionnaire on the Test Automation CSFs

Table 4.9 RII and Ranking of Test Automation CSFs

Test Automation	**RII**	**Rank**
42. Continuous integration testing by using automated testing is important for agile success automated acceptance testing	**0,842**	**1**
40. Increasing test coverage by using automated testing is important for agile success	**0,836**	**2**
44. Regression testing using automated testing is important for agile success	**0,831**	**3**
41. Test-driven development by using automated testing is important for agile success	0,796	4
43. Acceptance testing using automated testing is important for agile success	0,793	5

. The most influential factors are Continuous integration testing by using automated testing is important for agile success automated acceptance testing (RII = 0.842), Increasing test coverage by using automated testing is important for agile success (RII = 0.836) and Regression testing using automated testing is important for agile success (0.831). The least important factor was Acceptance testing using automated testing is important for agile success (RII = 0.793).

4.3.2.7 Cloud Computing

Table 4.10 shows the RII and ranking of the five statements from the questionnaire on the Cloud Computing CSFs

Table 4.10 RII and Ranking of Cloud Computing CSFs

Cloud Computing	**RII**	**Rank**
48. Automation of the end to end process (build, test, run) using cloud computing is important for agile success	**0,702**	**1**
49. User tool integration using cloud computing is important for agile success	**0,693**	**2**
47. Team collaboration using cloud computing is important for agile success	**0,682**	**3**
45. Infrastructure as a Service provided by cloud computing is important for agile success	0,669	4
46. Testing using cloud computing is important for agile success	0,644	5

The most influential factors for agile success are the automation of the end-to-end process (build, test, run) using cloud computing (RII = 0.702), User tool integration using the cloud (RII = 0.693), and Team collaboration using cloud computing (0.682). The least important factor was Testing using cloud computing for agile success (RII = 0.644).

4.4 OVERALL RANKING OF CSFS

Table 4.11 shows the mean of the RII and ranking of the seven CSFs in order of their importance as perceived by the respondents.

Table 4.11 RII showing CSFs ranked by Mean RII on Perceived Importance

CSFs	RII	RII %	Rank
People	0,8690	87%	**1**
Technical Factors	0,8384	84%	**2**
Test Automation	0,8196	82%	**3**
Organisational Structure	0,8093	81%	**4**
Process	0,8021	80%	**5**
Project	0,7535	75%	**6**
Cloud Computing	0,6778	68%	**7**

. The people factor (1^{st}) has the highest ranking as the RII of 0.869 is closest to 1 (87%), followed by the technical factors of 0.8384 (84%), test automation factor of 0.8196 (82%), the organisational structure of 0.8093 (81%), process factor of 0.8021 (80%), project factor of 0.7535 (75%) and the cloud computing factor of 0.6778 (68%) which is last (7^{th}) in the ranking.

4.5 DESCRIPTIVE ANALYSIS OF THE PERFORMANCE OF SUCCESS FACTORS IN AGILE PROJECTS

4.5.1 Organisational Structure

The following figure shows the results obtained for each of the ten statements that were presented on performance factor of organisational structure. The purpose was to analyze the responses received which indicated the performance of the organisational structure as perceived by the respondents.

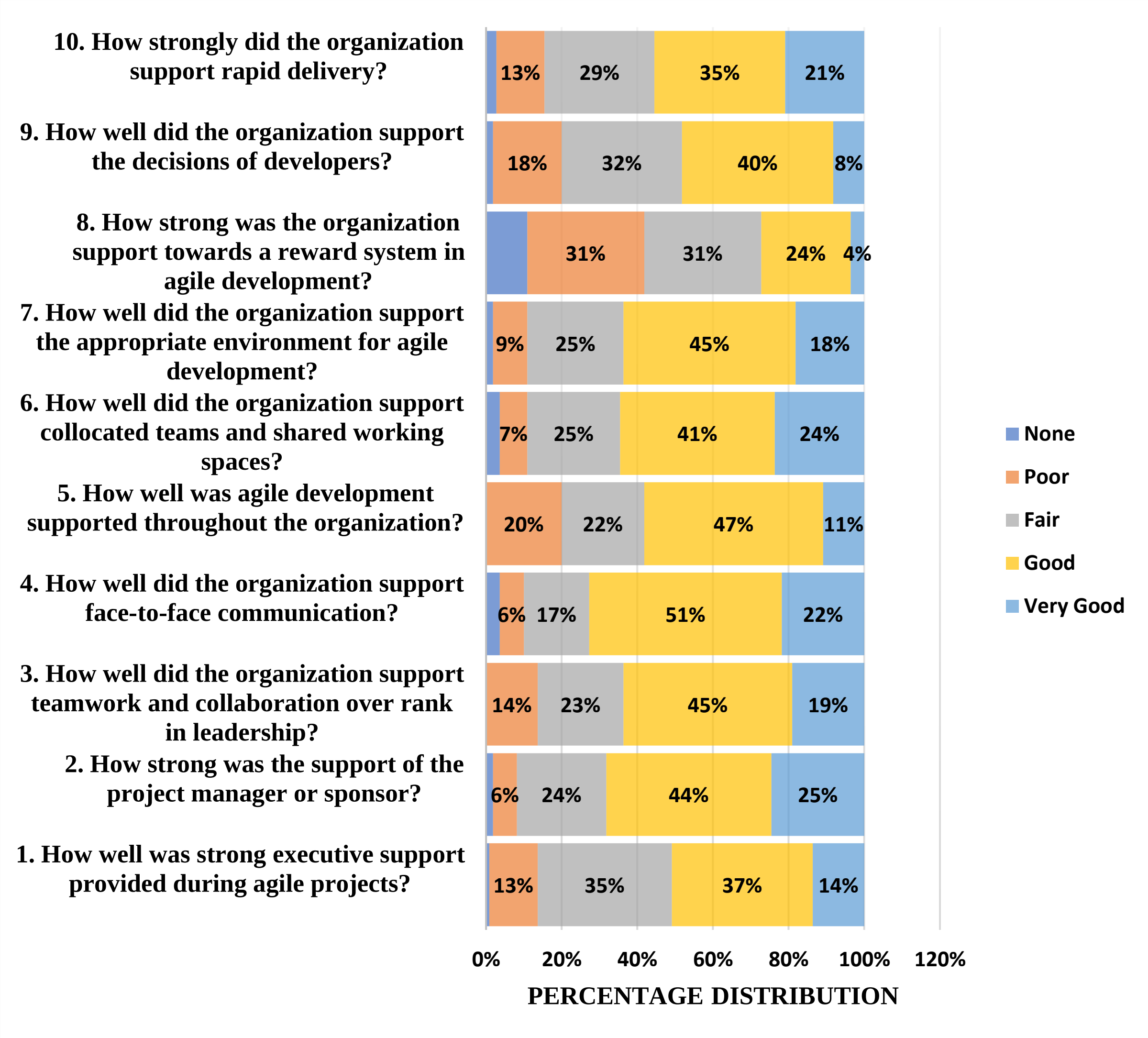

Figure 4.8 Descriptive Statistics for Organisational Structure

Figure 4.8 indicates that 73% of the respondents perceived that the organisation supported good to very good face-face communication, 69% responded that there was good to very good support provided by the project manager or sponsor, 65% indicated that there was good to very good support by the organisation to collocated teams and shared working spaces and 64% suggested there was good to very good support by the organisation for teamwork and collaboration over the rank in leadership. Although all the items for this factor were perceived to have good support from the organisation, the items described above show the items best supported by the organisation. A mean value of only 3.55, indicates that overall the respondents believed that the organisation provided marginally good support for this factor.

4.5.2 People

The following figure shows the results obtained for each of the seven statements that were presented on the performance factor of people. The purpose was to analyze the responses received which indicated the performance of the people factor for agile project success as perceived by the respondents.

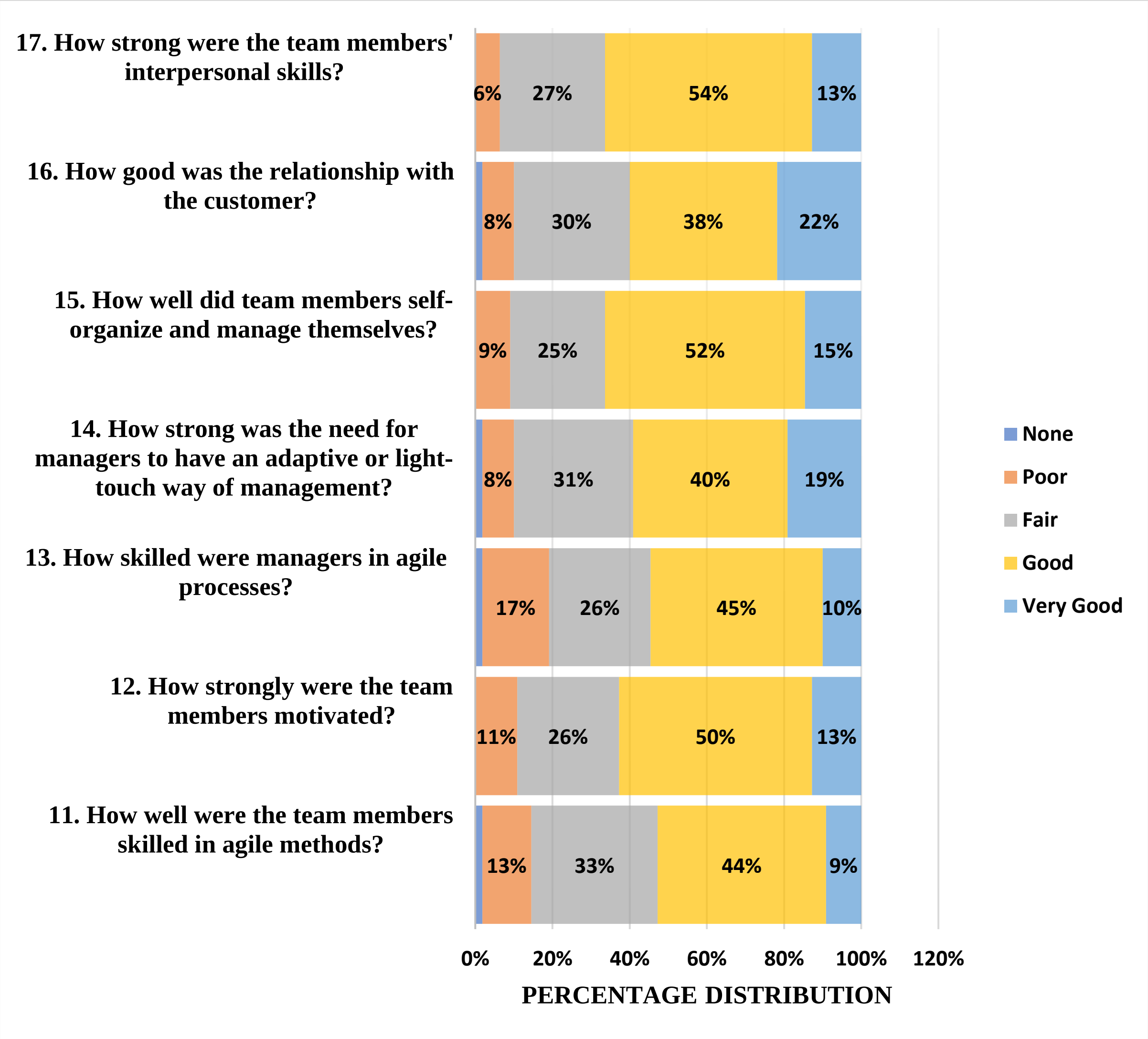

Figure 4.9 Descriptive Statistics for People

Figure 4.9 shows that 67% of respondents indicated that team members were good to very good in their ability to self-organize and manage themselves and 67% specified that team members had good to very good strong interpersonal skills, 60% perceived that the team members had good to very good motivation and 60% indicated that the agile stakeholders had a good to a very good relationship with the customer. Although all the items for the people factor were perceived by the respondents to be implemented and performed from good to very good, the items described above show the items implemented the best. A mean value of only 3.62, signifies that although the majority of the agile practitioners believed that the people factor was performing good to very good, many respondents indicated that some items performed from fair to good.

4.5.3 Process

The following figure shows the results obtained for each of the six statements that were presented on the performance factor of process. The purpose was to analyze the responses received which indicated the performance of the process factor for agile project success as perceived by the respondents.

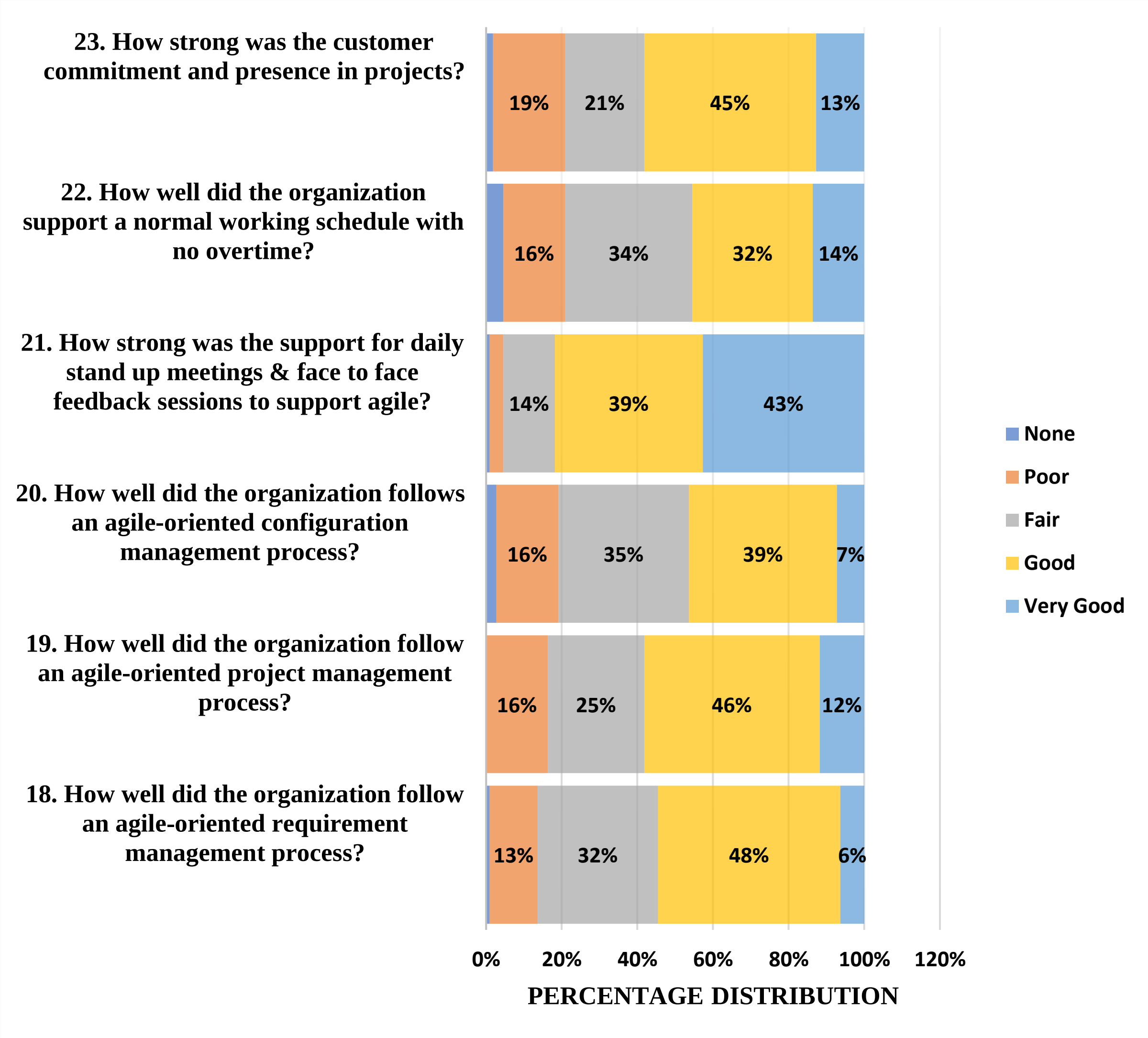

Figure 4.10 Descriptive Statistics for Process

Figure 4.10 indicates that 82% of the respondents indicated that there was good to very good support for daily stand-up meetings & face to face feedback sessions to support agile, 58% perceived that the organisation followed an agile-oriented project management process good to very good, 58% also indicated that customer commitment and presence in projects were good to very good and 54% indicated that the organisation was good to very good in following a strong agile-oriented requirement management process.

A mean value of only 3.55, indicates that overall the respondents believed that the organisation provided marginally good support for this factor.

4.5.4 Technical Factors

The following figure shows the results obtained for each of the nine statements that were presented on the performance factor of the technical factor. The purpose was to analyze the responses received which indicated the performance of the technical factor for agile project success as perceived by the respondents.

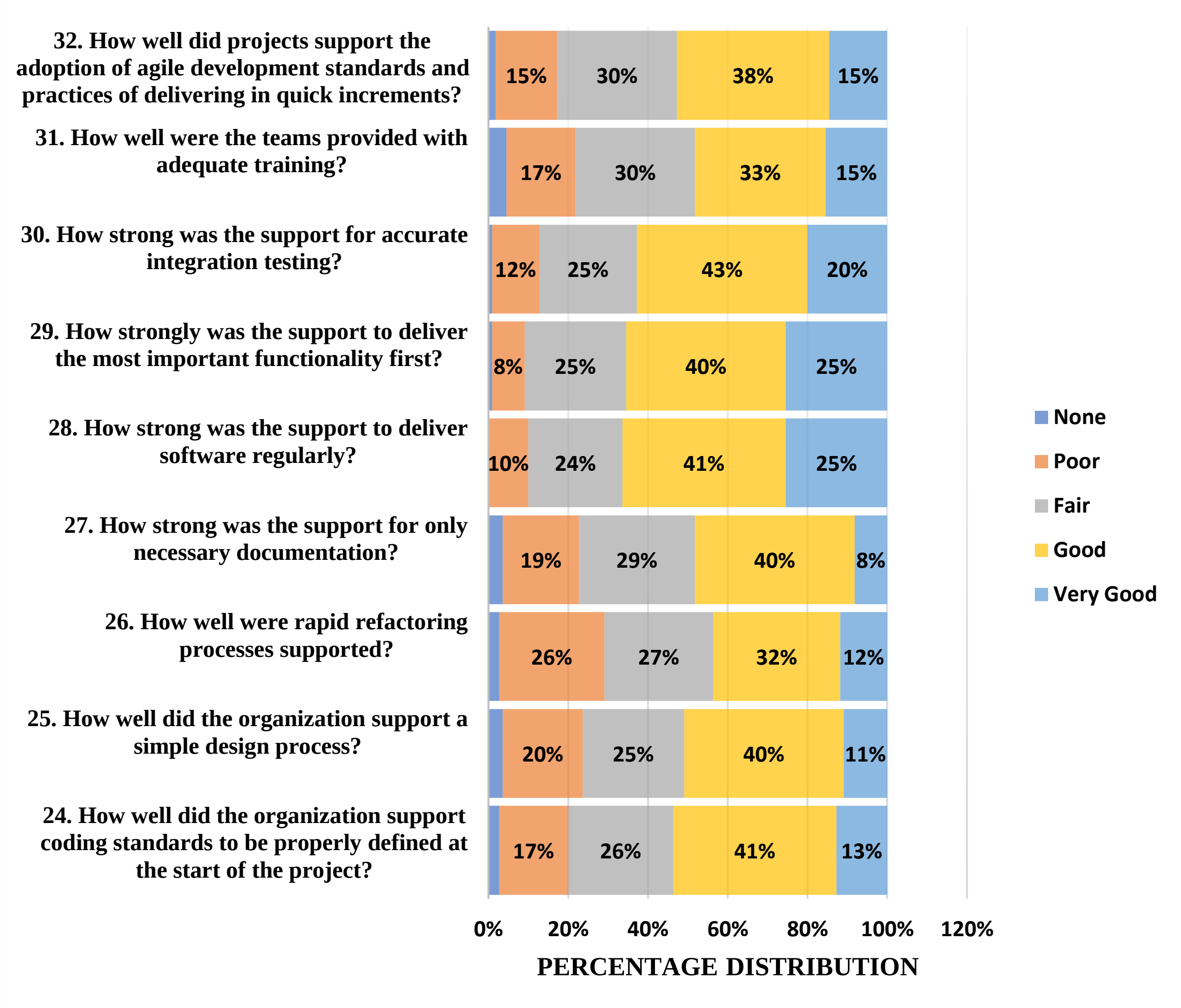

Figure 4.11 Descriptive Statistics for Technical Factors

Figure 4.11 indicates that 66% of respondents indicated that the support to deliver software regularly was good to very good, 65% mentioned that the support to deliver the most important functionality first was good to very good, 63% indicated that support for accurate integration testing was good to very good and 54% indicated that there was good to very good organisation support for coding standards to be properly defined at the start of the project. This is further substantiated by a mean value of 3.50 for this factor which shows that the agile project teams were marginally good in the implementation and performance of technical factors.

4.5.5 Project

The following figure shows the results obtained for each of the seven statements that were presented on the performance factor of project. The purpose was to analyze the responses received which indicated the performance of the project factor in agile project success as perceived by the respondents.

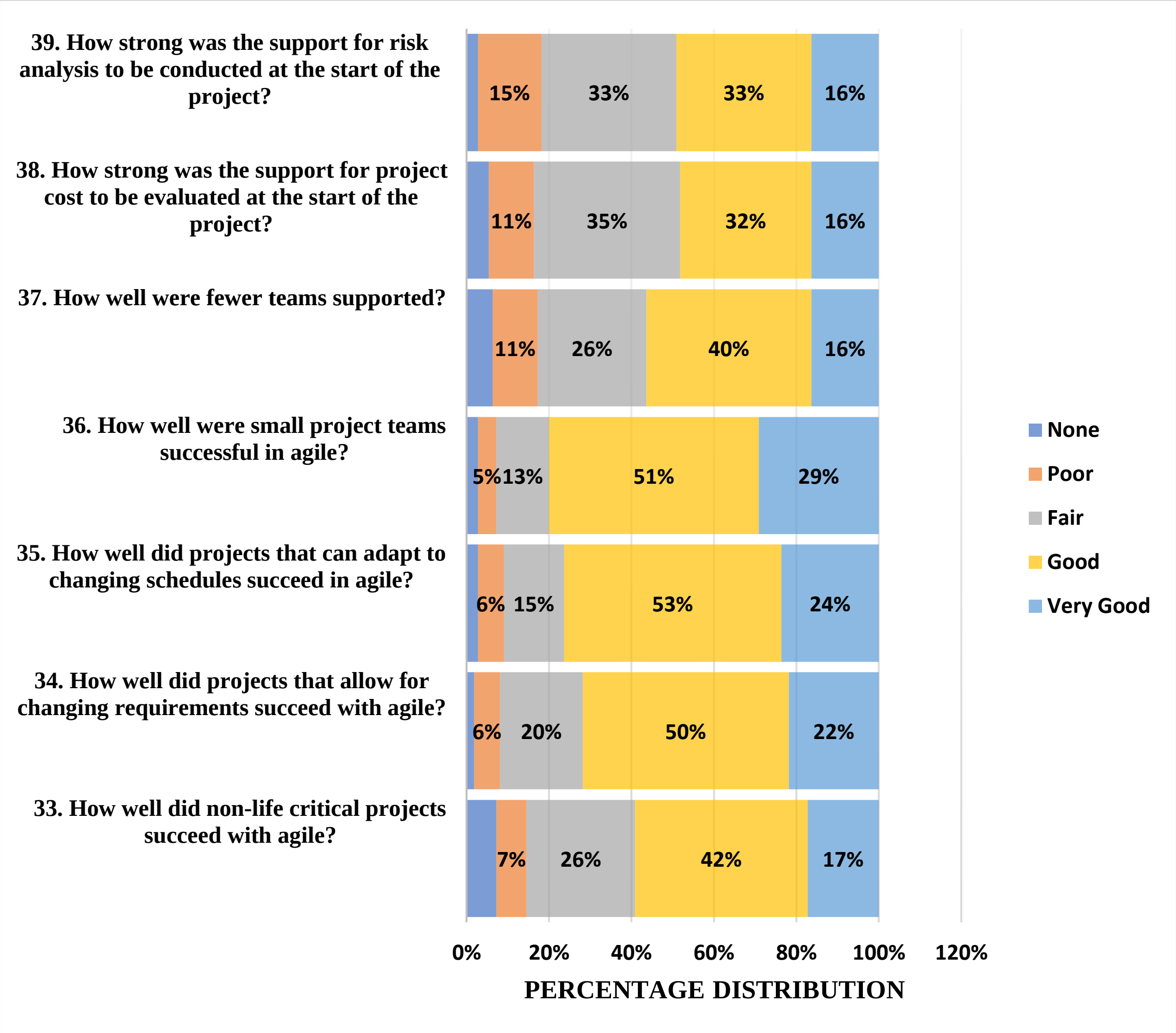

Figure 4.12 Descriptive Statistics for Project

Figure 4.12 indicates that 80% of respondents mentioned that small project teams were good to very good in being successful in agile, 77% indicated that projects were good to very good in adapting to changing schedules succeed well in agile, 72% mentioned that projects were good to very good in allowing success in agile with changing requirements, and 59% indicated that non-life critical projects were good to very good in succeeding well with agile. This is further substantiated by a mean value of 3.66 for this factor which indicates the marginally good performance of the project factor.

4.5.6 Test Automation

The following figure shows the results obtained for each of the five statements that were presented on test automation as a performance factor in agile project success. The purpose was to analyze the responses received which indicated the performance of test automation as perceived by the respondents.

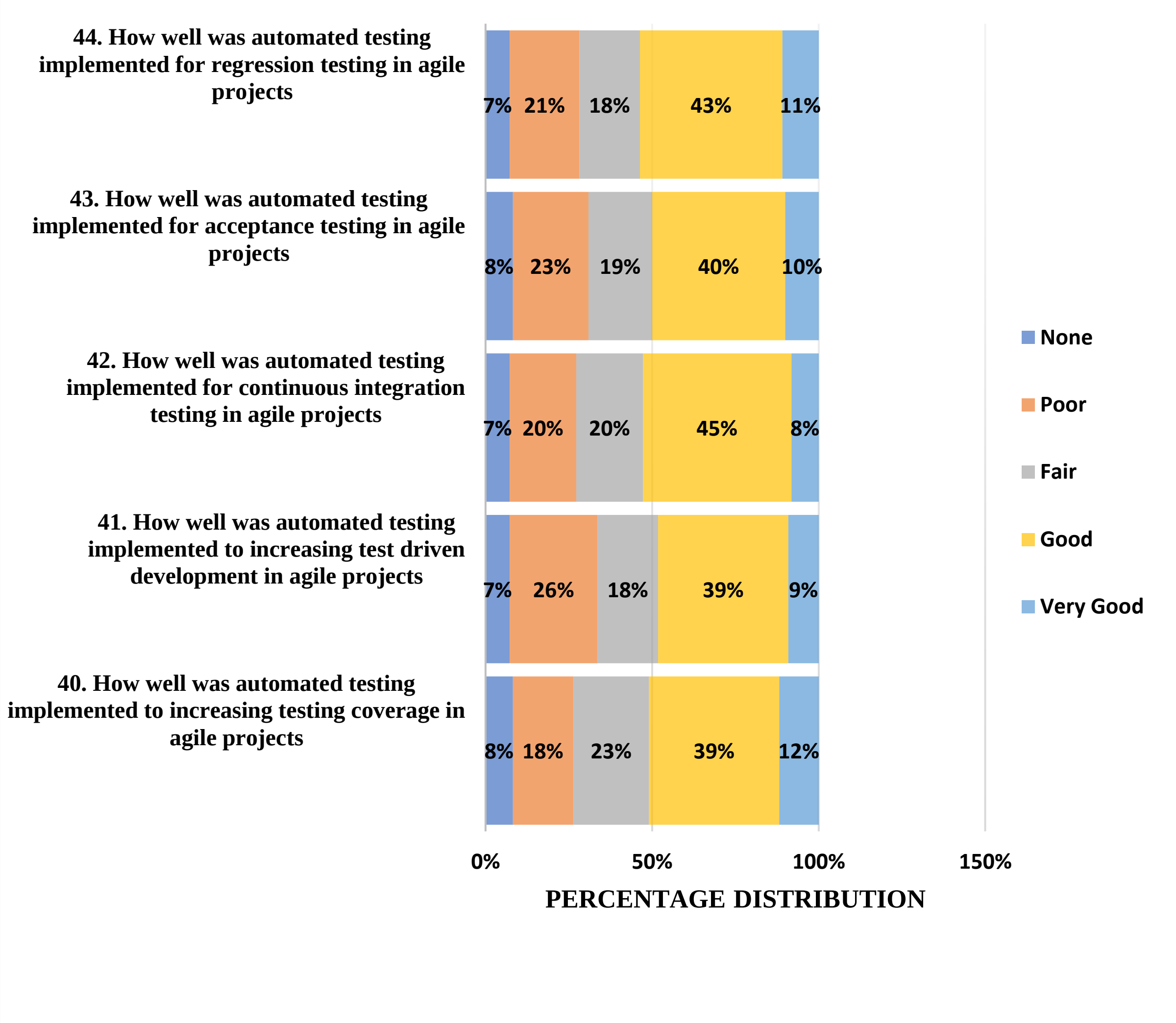

Figure 4.13 Descriptive Statistics for Test Automation

Figure 4.13 indicates that 54% of respondents indicate that there was good to very good implementation of automated testing for regression testing in agile projects, 53% mentioned that automated testing was implemented good to very good for continuous integration testing in agile projects and 54% indicated that automated testing was implemented good to very good to increase testing coverage in agile projects. A mean value of 3.24 for this factor indicates the performance of this factor was just above fair.

4.5.7 Cloud Computing

The following figure shows the results obtained for each of the five statements that were presented on the performance factor of cloud computing in the success of agile projects. The purpose was to analyze the responses received which indicated the performance of the cloud computing factor as perceived by the respondents.

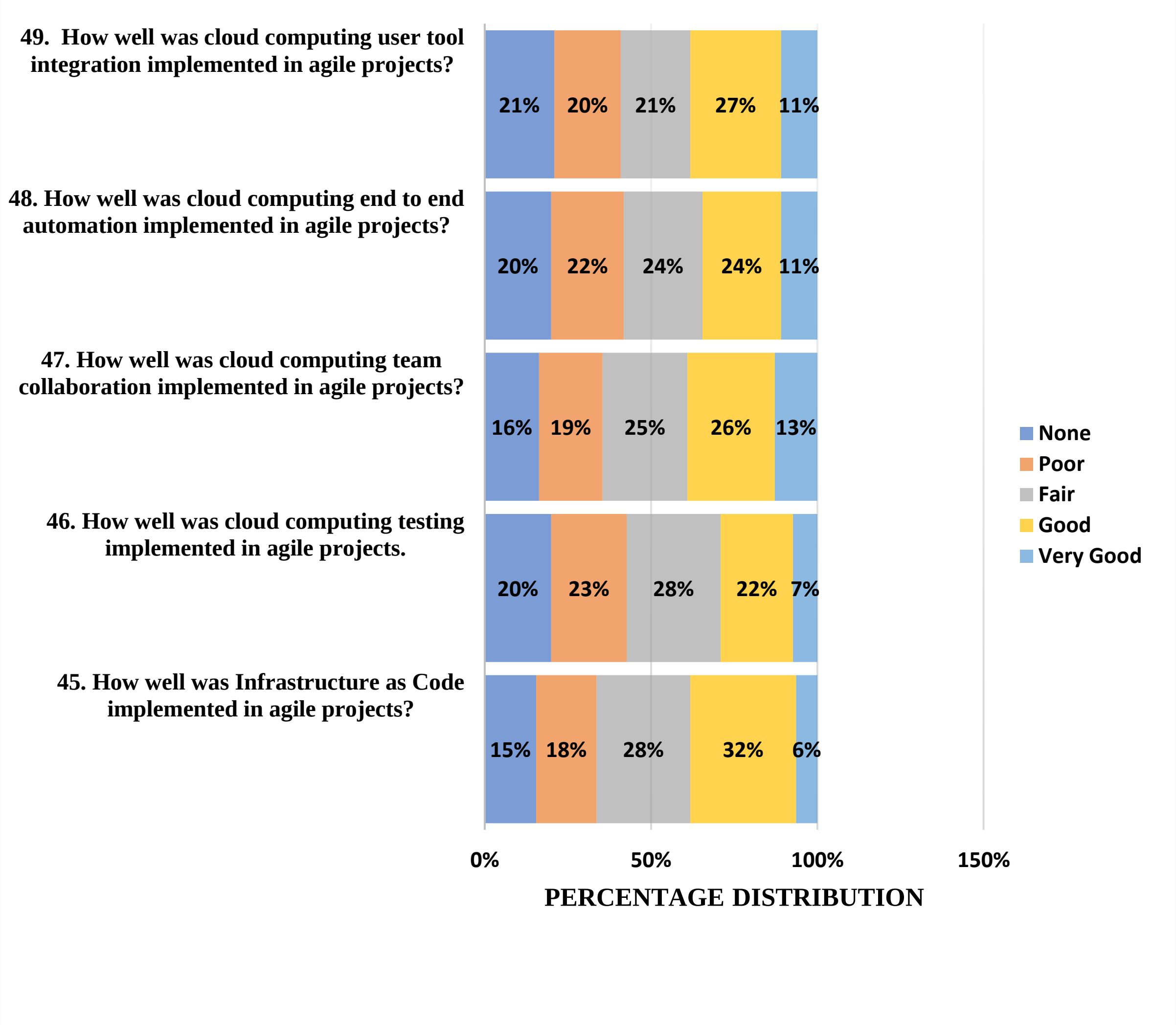

Figure 4.14 Descriptive Statistics for Cloud Computing

Figure 4.14 indicates that 39% of the respondents indicated that cloud computing team collaboration was implemented good to very good in agile projects, 38% suggested that Infrastructure as Code was implemented good to very good in agile projects and 38% suggested that cloud computing user tool integration was implemented good to very good in agile projects. The mean for each factor was three and below which indicates poor to fair performance of the cloud computing factor. This is further substantiated by a mean value of 2.88 for this factor.

The descriptive analysis indicated that certain factors are perceived by agile stakeholders to be have been implemented better or performed better than other factors during an agile project. Further, the items within each factor also showed varying degrees of performance or implementation. The next section shows the ranking of these factors and items using the relative importance index.

4.6 RII AND RANKING OF PERFORMANCE FACTORS

4.6.1 Organisational Structure

Table 4.12 shows the RII and ranking of the ten statements from the questionnaire on the performance of the Organisation Structure CSFs

Table 4.12 RII and Ranking of Organisation Structure CSFs

Organisational structure	**RII**	**Rank**
2. How strong was the support of the project manager or sponsor?	**0,765**	**1**
4. How well did the organisation support face-to-face communication?	**0,762**	**2**
6. How well did the organisation support collocated teams and shared working spaces?	**0,747**	**3**
3. How well did the organisation support teamwork and collaboration over the rank in leadership?	0,738	4
7. How well did the organisation support the appropriate environment for agile development?	0,738	4
10. How strongly did the organisation support rapid delivery?	0,716	5
1. How well was strong executive support provided during agile projects?	0,700	6
5. How well was agile development supported throughout the organisation?	0,698	7
9. How well did the organisation support the decisions of developers?	0,669	8

8. How strong was the organisation's support towards a reward system in agile development?	0,556	9

The items that had the highest performance within this factor are: agile stakeholders perceived that there was strong support by the project managers or sponsors (RII = 0.765), the organisations supported face-to-face communication (RII = 0.762), there was strong executive support provided during agile projects (RII = 0.700), the organisation supported collocated teams and shared working spaces (RII = 0.747), the organisation supported teamwork and collaboration over the rank in leadership (RII = 0.738) and the organisation supported the appropriate environment for agile development (0.738). The least performing factor was the organisation's support of a reward system in agile development (RII = 0.556).

4.6.2 People

Table 4.13 shows the RII and ranking of the seven statements from the questionnaire on the performance of the People CSFs

Table 4.13 Performance: RII and Ranking of People CSFs

People	RII	Rank
17. How strong were the team members' interpersonal skills?	**0,745**	**1**
15. How well did team members self-organize and manage themselves?	**0,744**	**2**
16. How good was the relationship with the customer?	**0,740**	**3**
14. How strong was the need for managers to have an adaptive or light-touch way of management?	0,733	4
12. How strongly were the team members motivated?	0,729	5
11. How well were the team members skilled in agile methods?	0,691	6
13. How skilled were managers in agile processes?	0,687	7

The most influential factors were: the team members' interpersonal skills were strong (RII = 0.745), the team members were self-organized and managed themselves well (RII = 0.744), the relationship was good with the customer (RII = 0.740), there was a strong need for managers to have an adaptive or light-touch way of management (RII = 0.733) and the team members were strongly motivated (RII = 0.729). The least performing factor was the skill level of managers in agile processes? (RII = 0.687).

4.6.3 Process

Table 4.14 shows the RII and ranking of the six statements from the questionnaire on the performance of the Process CSFs

Table 4.14 Performance: RII and Ranking of Process Factor

Process	**RII**	**Rank**
21. How strong was the support for daily stand up meetings & face to face feedback sessions to support agile?	**0,838**	**1**
19. How well did the organisation follow an agile-oriented project management process?	**0,707**	**2**
23. How strong was customer commitment and presence in projects?	**0,696**	**3**
18. How well did the organisation follow an agile-oriented requirement management process?	0,693	4
22. How well did the organisation support a normal working schedule with no overtime?	0,667	5
20. How well did the organisation follows an agile-oriented configuration management process?	0,664	6

The most influential factors are strong support for daily stand-up meetings & face to face feedback sessions to support agile (RII = 0.838), the organisation followed an agile-oriented project management process (RII = 0.707), and customer commitment and presence in projects was strong (RII = 0.696). The least performing Process factor was the level at which the organisation followed an agile-oriented configuration management process (RII = 0.664).

4.6.4 Technical Factors

Table 4.15 shows the RII and ranking of the nine statements from the questionnaire on the performance of the Technical Factor CSFs

Table 4.15 Performance: RII and Ranking of Technical Factor CSFs

Technical Factors	RII	Rank
28. How strong was the support to deliver software regularly?	**0,764**	**1**
29. How strongly was the support to deliver the most important functionality first?	**0,762**	**2**
30. How strong was the support for accurate integration testing?	**0,738**	**3**
32. How well did projects support the adoption of agile development standards and practices of delivering in quick increments?	0,696	4
24. How well did the organisation support coding standards to be properly defined at the start of the project?	0,687	5
31. How well were the teams provided with adequate training?	0,675	6
25. How well did the organisation support a simple design process?	0,669	7
26. How well were rapid refactoring processes supported?	0,647	8
27. How strong was the support for only necessary documentation?	0,660	9

The most influential factors are the strong support to deliver software regularly (RII = 0.764), the strong support to deliver the most important functionality first (RII = 0.762), and the strong support for accurate integration testing (RII = 0.738). The least performing Technical factor was the support for only necessary documentation? (RII = 0.660).

4.6.5 Project

Table 4.16 shows the RII and ranking of the seven statements from the questionnaire on the performance of the Project CSFs

Table 4.16 Performance: RII and Ranking of Project CSFs

Project	RII	Rank
36. How well were small project teams succeed in agile?	**0,798**	**1**
35. How well did projects that can adapt to changing schedules succeed in agile?	**0,776**	**2**
34. How well did projects that allow for changing requirements succeed with agile?	**0,767**	**3**
33. How well did non-life critical projects succeed with agile?	0,709	4
37. How well were fewer teams supported?	0,698	5
38. How strong was the support for project cost to be evaluated at the start of the project?	0,685	7
39. How strong was the support for risk analysis to be conducted at the start of the project?	0,689	8

The most influential factors were small project teams successful in agile (RII = 0.798), projects that can adapt to changing schedules succeed in agile (RII = 0.776), and projects that allow for changing requirements succeed with agile (RII = 0.767). The least performing Project factor was the support for risk analysis to be conducted at the start of the project (RII = 0.689).

4.6.6 Test Automation

Table 4.17 shows the RII and ranking of the five statements from the questionnaire on the performance of the Test Automation CSFs

Table 4.17 Performance: RII and Ranking of Test Automation CSFs

Test Automation	RII	Rank
44. How well was automated testing implemented for regression testing in agile projects?	**0,658**	**1**
40. How well was automated testing implemented to increasing testing coverage in agile projects?	**0,656**	**2**

42. How well was automated testing implemented for continuous integration testing in agile projects?	**0,653**	**3**
43. How well was automated testing implemented for acceptance testing in agile projects?	0,642	4
41. How well was automated testing implemented to increasing test-driven development in agile projects?	0,633	5

Table 4.17 shows the RII and ranking of the Test Automation CSFs. The most influential factors were automated testing implemented for regression testing in agile projects (RII = 0.658), automated testing implemented to increase testing coverage in agile projects (RII = 0.656), and automated testing implemented for continuous integration testing in agile projects (RII = 0.653). The least performing Project factor was automated testing implemented to increase test-driven development in agile projects? (RII = 0.633).

4.6.7 Cloud Computing

Table 4.18 shows the RII and ranking of the five statements from the questionnaire on the performance of the Cloud Computing CSFs

Table 4.18 Performance: RII and Ranking of Cloud Computing CSFs

Cloud Computing	**RII**	**Rank**
47. How well was cloud computing team collaboration implemented in agile projects?	**0,600**	**1**
45. How well was Infrastructure as Code implemented in agile projects?	**0,591**	**2**
49. How well was cloud computing user tool integration implemented in agile projects?	**0,575**	**3**
48. How well was cloud computing end to end automation implemented in agile projects?	0,567	4
46. How well was cloud computing testing implemented in agile projects.	0,547	5

The most influential factors were cloud computing team collaboration implemented in agile projects (RII = 0.600), Infrastructure as Code implemented in agile projects (RII = 0.591), and cloud computing user tool integration implemented in agile projects (RII = 0.575). The least performing factor was cloud computing testing implemented in agile projects (RII = 0.547)

4.7 OVERALL RANKING OF PERFORMANCE CSFS

Table 4.19 shows the mean RII which indicates the ranking of the seven CSFs in terms of their perceived performance in agile projects by the respondents

Table 4.19 RII showing CSFs ranked by Mean RII on Perceived Performance

CSFs	**RII**	**RII %**	**Rank**
Project	0,7319	73%	**1**
People	0,7242	72%	**2**
Organisational Structure	0,7091	71%	**3**
Process	0,7109	71%	**4**
Technical Factors	0,6998	70%	**5**
Test Automation	0,6484	65%	**6**
Cloud Computing	0,5760	58%	**7**

. The project factor (1st) has the highest ranking as the RII of 0.7319 is closest to 1 (73%), followed by the people factor of 0.7242 (72%), organisational structure of 0.7091 (71%), process factor of 0.7109 (71%), technical factors of 0.6998 (70%), test automation factor of 0.6484 (65%) and the cloud computing factor of 0.5760 (58%) which is last (7th) in the ranking. The higher relative importance indicates that these are the highest performing factors in agile projects. It also takes into account the weighting of the responses received which are based on the Likert scale. In this way, the responses are not tallied only by their frequency, but rather for the weighting of each response received.

4.8 THE GAP BETWEEN THE PERCEIVED IMPORTANCE AND ACTUAL PERFORMANCE OF SUCCESS FACTORS

4.8.1 Reliability Analysis

Cronbach's Alpha was used to conduct a reliability analysis. **If Cronbach's coefficient is higher than 0.7, this implies that the result is reliable (Taber, 2018).**

Table 4.20 Cronbach - Reliability Analysis - Importance

Importance CSFs	**Cronbach's Alpha**	**Number of Items**
Organisational Structure	0,779	10
People	**0,877**	7
Process	0,734	6
Technical Factors	0,789	9
Project	0,622	7
Test Automation	**0,921**	5
Cloud Computing	**0,926**	5

Table 4.20 shows the results from Cronbach's Alpha test for the Importance factors are shown in Table above and show Cloud Computing with the highest Alpha value of 0.926, followed closely by Test Automation with a value of 0.921 and People with a value of 0.877. The generally accepted value of 0.7 for Cronbach's alpha is generally accepted as the benchmark, but values that are above 0.6 are accepted (Griethuijsen et al., 2015). The project CSF had the lowest value of 0.622 which is lower than the 0.7 benchmarks but is within the acceptable range of higher than 0.6. It can therefore be accepted as reliable.

Table 4.21 Cronbach - Reliability Analysis - Performance

Performance CSFs	**Cronbach's Alpha**	**Number of Items**
Organisational Structure	**0,877**	10
People	**0,812**	7
Process	**0,814**	6
Technical Factors	**0,899**	9
Project	0,789	7
Test Automation	**0,967**	5

Cloud Computing	**0,963**	5

Table 4.21 shows the results from Cronbach's Alpha test for the Performance factors are below and show Test Automation with the highest Alpha value of 0.967, followed closely by Cloud Computing with a value of 0.963 and Technical Factors with a value of 0.899. The project factor had the lowest value of 0.789 which is higher than the 0.7 benchmark which indicates the reliability of the data collected for the Performance factors (Griethuijsen et al., 2015).

4.8.2 Paired Sample T-Test:

The t-test was conducted to determine if there was a significant gap between the perceived importance and actual performance for each critical success factor. A T-Test is a type of inferential statistic that is used as a tool to tests hypotheses on data that follows a normal distribution (Kenton, 2020). It is used to investigate whether the difference between the means of two samples or groups is significant (Siegle, 2020). Statistical significance is the likelihood of obtaining a deviation from the null hypothesis and is referred to as the p-value or probability value. The null hypothesis is rejected if $p < 0.05$.

The central limit theorem suggests that regardless of the distribution of observations, large samples will follow a normal distribution (Altman & Bland, 1995). This theory states that if the sample size is 100 or more observations, then the data will follow a normal distribution (Altman & Bland, 1995). This dismisses the need to conduct normality tests for this research as the assumption is that it will follow a normal distribution. Therefore, the paired sample t-test was performed on the data collected. A **95% confidence level** adopted by the paired sample T-test indicates that if 100 samples produce 100 sample means from a population, **95 are included in the confidence interval and 5 will be out of this interval** (Lee, In & Lee, 2015).

Table 4.22 Paired Samples T-Test

	Factor	**Mean Importance**	**Mean Performance**	**Paired Difference**	**t**	**df**	**Sig. (2-tailed)**
Pair 1	**Organisational Structure**	4,05	3,55	0,5	**6,702**	**109**	**0.0001**
Pair 2	**People**	4,35	3,62	0,72	**11,594**	**109**	**0.0001**

Pair 3	**Process**	4,01	3,55	0,46	**5,923**	**109**	**0.0001**
Pair 4	**Technical**	4,19	3,50	0,69	**10,073**	**109**	**0.0001**
Pair 5	**Project**	3,77	3,66	0,11	**1,773**	**109**	**0,079**
Pair 6	**Test Automation**	4,10	3,24	0,86	**8,377**	**109**	**0.0001**
Pair 7	**Cloud Computing**	3,39	2,88	0,51	**4,835**	**109**	**0.0001**

Table 4.22 shows the following:

- **Pair 1. Organisational Structure** - The means for perceived importance and perceived performance *do* differ statistically significantly, $t(109) = 6.702$, $p < 0.001$ i.e. the Importance and Performance factors do differ significantly.
- **Pair 2. People -** The means for perceived importance and perceived performance *do* differ statistically significantly, $t(109) = 11.594$, $p < 0.001$ i.e. the Importance and Performance factors do differ significantly.
- **Pair 3. Process** - The means for perceived importance and perceived performance *do* differ statistically significantly, $t(109) = 5.923$, $p < 0.001$ i.e. the Importance and Performance factors do differ significantly.
- **Pair 4. Technical Factors** - The means for perceived importance and perceived performance *do* differ statistically significantly, $t(109) = 10.073$, $p < 0.001$ i.e. the Importance and Performance factors do differ significantly.
- **Pair 5. Project** - The means for perceived importance and perceived performance *do* not differ statistically significantly, $t(109) = 1.773$, $p > 0.05$
- i.e. the Importance and Performance factors do not differ significantly.
- **Pair 6. Test Automation -** The means for perceived importance and perceived performance *do* differ statistically significantly, $t(109) = 8.377$, $p < 0.001$ i.e. the Importance and Performance factors do differ significantly.
- **Pair 7. Cloud Computing** - The means for perceived importance and perceived performance *do* differ statistically significantly, $t(109) = 4.835$, $p < 0.001$ i.e. the Importance and Performance factors do differ significantly.

Table 4.23 Summary of Hypothesis Testing for CSFs

Hypothesis	Relationship Tested	Result
H1a (Pair 1)	There is a significant difference between the importance of Organisational Structure and its performance	Accepted $p < 0.001$
H1b (Pair 2)	There is a significant difference between the importance of People and their performance	Accepted $p < 0.001$
H1c (Pair 3)	There is a significant difference between the importance of Process and its performance	Accepted $p < 0.001$
H1d (Pair 4)	There is a significant difference between the importance of Technical Factors and their performance	Accepted $p < 0.001$
H1e (Pair 5)	There is no significant difference between the importance of the Project and its performance	Rejected $p > 0.05$
H1f (Pair 6)	There is a significant difference between the importance of Test Automation and its performance	Accepted $p < 0.001$
H1g (Pair 7)	There is a significant difference between the importance of Cloud Computing and its performance	Accepted $p < 0.001$

Table 4.23 presents a summary of the results obtained for the hypothesis testing. The result column shows that $p< 0.001$ for each pairing of critical success factor for importance and its corresponding performance factor, except for the project factor where $p>0.05$.

It indicates that the null hypothesis is rejected, and the alternative hypothesis is accepted for the factors of organisational structure, people, process, technical factors, test automation, and cloud computing as $p<0.001$. These results indicate that the difference between the importance and performance factors was statistically significant for these factors. The null hypothesis is only supported for the project factor as $p>0.05$ and this indicates that there is no significant difference between the perceived importance and perceived performance of the project factor.

Further, the project factor was the highest-ranking performance factor which indicates that this factor performs well in the organisation's projects. The importance factors were to measure whether a factor was important for agile success and performance was to determine if these factors performed well in achieving success if they were implemented. A gap indicates that although the stakeholders think the factor is important, this factor did not perform well in helping to achieve project success. The organisation for example needs to put strategies in place to ensure that these factors are performing well to ensure agile success. The gap was not significant for the project factor as agile projects generally allow for changing requirements and therefore the response to the project's importance and performance was balanced. The highest-ranking importance factors were people, technical factors, and test automation. The highest-ranking performance factors were the project, people, and organisational structure. The people factor was the only factor that was common in the top three importance and performance ranking.

The bottom three importance factors were process, project, and cloud computing. Technical factors, test automation, and cloud computing were the bottom three performing factors. Technical factors and test automation were also in the top three importance factors which indicate that although these two factors are important, they do not perform well in agile projects. Cloud computing was the only factor that was common in the bottom three importance and performance ranking.

4.9 CONCLUSION

This chapter presents the data that was collected using a closed-ended questionnaire. The research data was analyzed, interpreted, and reported in the form of tables and graphs. The biographical data was presented and analyzed by descriptive statistics to assist in understanding the respondents and their attributes. It assists in adding context to the statistics by expressing the population dynamics and other characteristics. The Relative Importance Index test is then calculated by computing the mean of the Importance and Performance factors via SPSS and thereafter the RII formula was used to establish the ranking of the CSFs. The top three importance factors were people, technical factors, and test automation, while the top three performance factors were the project, people, and organisational structure. The bottom three importance factors were process, project, and cloud computing, while the bottom three performance factors were technical, test automation, and cloud computing factors. It was noted that the people factor featured in the top three importance and performance ranking, while only cloud computing featured in the bottom three importance and performance ranking. Paired Samples T-Test was used to determine the gap between the importance and performance of the CSFs. The gap was significant for all the CSFs with the exclusion of the project factor. The discussion of the results follows in chapter 5.

CHAPTER 5

DISCUSSION, RECOMMENDATIONS, AND CONCLUSION

5.1 INTRODUCTION

The goal of this research was to investigate the critical success factors that affect the success of agile software projects. The emphasis was to ascertain the hierarchy of the importance of the critical success factors and their performance in projects. Literature from previous research was evaluated, together with the findings from the questionnaire.

This chapter entails a discussion of the results obtained from the data analysis in the previous chapter, recommendations, the limitations of the research study, possible future research, and conclusions that were established. It considers the findings and contextualizes the concepts into the real world via the recommendations.

5.2 DISCUSSION

Research Q1: Which critical factors are perceived to be most important by stakeholders for agile success?

The findings from the mean relative importance index in chapter 4 revealed the ranking of the critical success factors in terms of their importance. The people factor ranked the highest which reveals that the respondents viewed agile skills, interpersonal skills, the characteristics of managers and team members, and the relationship with the customer as being the most important critical success factors for agile success. The technical factor ranked second which revealed the importance of coding standards, technical training, design specification, and alignment to agile practices. The test automation factor was ranked third due to the respondents' view on continuous regression testing, increased test coverage, and acceptance testing using automated testing. The cloud computing factor ranked 7th (last), and this was due to the respondents' negative view on testing using cloud computing, infrastructure as a service provided by cloud computing, and team collaboration using cloud computing.

In a similar study by Chow and Cao (2008), team members not having the appropriate agile skill set, no teamwork, and less cooperation from team members were found to be factors that contribute to the failure of agile projects. Chow and Cao (2008) found that the factor that was considered to have the greatest impact on the success of agile projects was the technical factors, followed by the people factor. Chow and Cao's (2008) study also aligns with this research as in this research people and technical factors were the two most critical success factors.

In another similar study by Islam (2016) technical factors were the most critical success factor, followed by process factors, organisational factors, people factors, and project factors. Technical factors were ranked second amongst the critical success factors in this study which aligns with the study by Islam (2016) where it was ranked first. However, people factors were ranked fourth compared to being the most critical success factor in this research. This does not align with this study and may be due to the data collection method of Islam's (2016) study. His research included analyzing 31 research studies (24 accepted, 7 rejected) and a frequency analysis which was conducted on the data that was extracted. This could be the reason people were viewed as the fourth as compared to the first in this study. Further Islam's (2016) study used secondary data from previous literature, whereas primary data was collected in this research. The usage of secondary data could have influenced the CSFs in Islam's (2016) study.

In another similar study by Misra et al. (2009), nine out of fourteen possible factors were linked to success. They are customer satisfaction, customer collaboration, customer commitment, decision time, corporate culture, control, personal characteristics, societal culture, and training and learning. These factors reflect the people component that is being investigated and emphasizes the findings of this study where the people factor is ranked highest for agile success.

Research Q2: What is the performance of these CSFs in the organisation?

The results indicate that the critical success factors of organisational structure, people, process, technical factors, project, test automation, and cloud computing have means below 4 which show that respondents view these CSFs as having moderate performance and the cloud computing factor as having poor performance (below 3). The results indicate that these critical success factors were not performing well in agile projects. Although the results indicated the importance of the critical success factors in research question 1 above, the perceived performance of these CSFs is low and questionable. The highest performing individual factor was the process factor of the strong support for daily stand-up meetings and feedback sessions. An example of the people factor was the support for team members' interpersonal skills which performed well. The project factors example, smaller project teams and teams that can adapt to changing requirements performed well. The technical factor of the support to deliver software regularly also performed well. The worst performing individual factors were the cloud computing factor of implementing cloud computing testing and the organisational factor of a reward system in agile projects.

Research Q3: What is the gap between the perceived importance and the actual performance of the CSFs?

The results indicate that there is a gap between the importance and performance of the critical success factors. This shows that although the respondents perceive that the critical success factors are important in achieving success in agile projects, this is not exercised in practice in projects. The highest gap was between the test automation factor which was closest to 1 which is indicative that this factor is lacking in its performance. This could also be as a result of the organisation not perceiving this factor to be important and hence not investing resources in its performance. This was followed by the difference between the people and technical factors. Organisations must focus on the performance of the critical success factors to ensure success in agile projects. The gap was the highest among the top three critical success factors of people, technical factors, and test automation. This is indicative of the reality of agile projects where the performance of the most perceived critical factors has the highest gap in performance. People are viewed as the most critical factor yet the gap is second highest, technical factors are viewed as the second most critical factor yet the gap is third-highest and test automation which is viewed as the third most critical factor has the highest gap. This is indicative of the gap that exists between the importance and the performance of the critical success factors. Factors are important but are not performing as well in organisations.

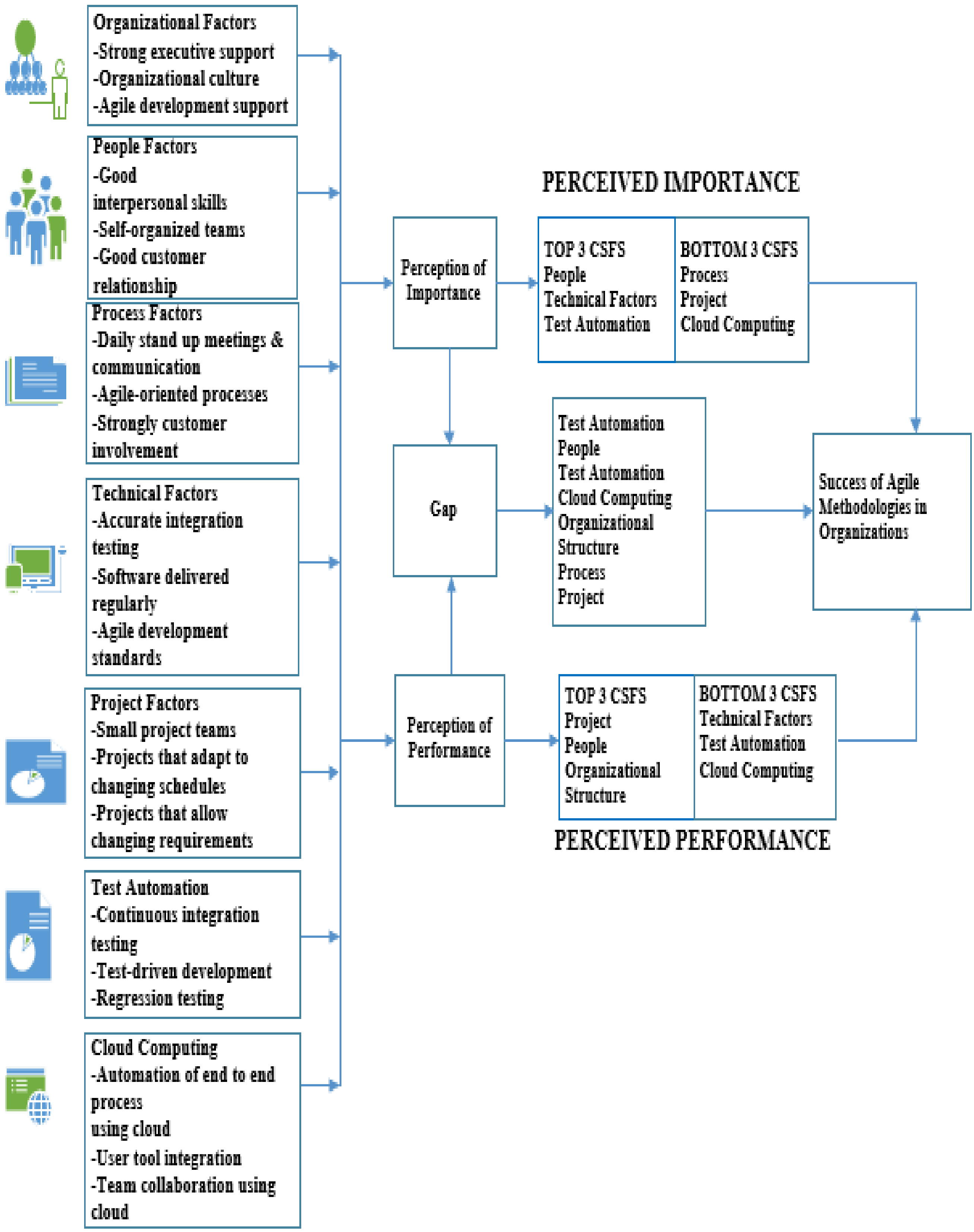

Figure 5.1 Research framework of critical success factors

The final research framework is shown in Figure 5.1. The top and bottom three critical success factors were identified in terms of perceived importance and perceived performance. The top three CSFs of importance are people, technical factors, and test automation, while process projects and cloud computing formed the bottom three. The top 3 performing CSFs are project, people, and organisational structure, while technical factors, test automation, and cloud computing formed the bottom three. This ranking was not evident in the initial framework of the research. It is noted that technical factors and test automation form the top 3 in terms of importance. The project is in the top three in terms of performance but in the bottom three of importance. This shows that the project performs well but is not perceived as the most important CSFs. Cloud computing was in the bottom three for importance and performance. People are featured in the top three for both importance and performance, which shows that people are perceived as an important factor and this factor performs in practice as well.

5.3 RECOMMENDATIONS

Organisations should assign priority to the critical success factors that affect agile projects. The ranking of the critical success factors will assist organisations to place greater emphasis on the success of these factors which will, in turn, result in successful agile implementation. The top three critical success factors identified were People, Technical Factors, and Test Automation. Respondents ranked people as the top critical success factor which is indicative of the importance of this factor. Organisations should focus on the upskilling of their project teams in agile methods, the necessary technical training, interpersonal skills, and the relationship with the managers and customers. The people factor is integral to agile project success and the recommendation of this research is to focus on the motivation of team members, the agile skills of the team including managers, interpersonal skills, and the relationship with each other and the customer. Organisations should focus on these people factors and the upskilling of their project teams in agile technologies. The results showed that respondents agreed that those team members who are strongly motivated contribute to agile success and this shows the importance of people as a critical success factor. Organisations spend exorbitant amounts of money on technical resources like outsourced personnel, equipment, software packages, office space etcetera, but little focus is on the people who drive projects.

Technical factors that ranked second showed that respondents valued software that was delivered regularly, adoption of agile standards, and agile technical training for agile success. The researcher's recommendation on technical factors is for organisations to invest in adequate training for their team members and the implementation of agile standards in practice. Further, organisations should have short sprints to ensure that functionality is delivered regularly so that team members are fulfilled in the output that they deliver. Test automation ranked third and suggest the respondents' view that automated testing was important for agile success. The researcher's recommendation is for organisations to consider automated testing in agile projects as this was indicated as a critical success factor. Automated testing will eliminate the need for manual testing and will ensure a quicker turnaround for testing. Further, it will eliminate errors that could result from human error in testing. The overall recommendation is for organisations to focus on the critical success factors as identified in this research, but to pay special attention to the hierarchy of importance of these factors. Further, the findings revealed that there is a definitive disjuncture between the perceived importance and perceived performance of these critical success factors. The second recommendation is for organisations to ensure that these critical success factors are not only given importance, but their performance should be monitored to ensure agile project success. The performance should be monitored by the implementation of key performance indicators which will allow organisations to audit the performance of the CSFs. Project teams will be measured on these KPIs and will therefore be inclined and motivated to ensure the good performance of the CSFs. Organisations must focus on the critical success factors identified for agile projects to succeed with a particular emphasis on the top three identified factors of people, technical factors, and test automation.

For organisations that want to adopt agile methodologies, they should primarily focus on the critical success factors. These factors should be performing well in the organisation to ensure agile success. The critical success factors and their ranking in terms of importance were identified in this research which organisations should use to direct their efforts towards. The top three critical success factors identified were the people factor, the technical factor, and the test automation factor. Therefore, organisations adopting agile methodologies should concentrate on these factors to ensure agile success. Although there are other critical factors, organisations should expend extra effort on the top three critical factors.

To narrow the gap between the importance and performance factors, organisations should motivate their project team members to focus on the important factors and strive to encourage the project members to guide the performance of these critical success factors. For example, since the people factor was the highest-ranking importance factor, organisations should plan for the training of the project team members, upskilling of interpersonal skills, and encourage the relationship with the customer and with the other team members.

5.4 LIMITATION OF THE STUDY AND FUTURE RESEARCH

The study was limited to South African agile practitioners and is therefore limited to the South African agile projects. The study cannot be fully generalized on a global scale.

The stakeholders in projects are not limited to software developers and the perception of management, analysts, and other personnel are important as members of project teams. The researcher did attempt to source most of the agile project team members.

Only participants who had LinkedIn accounts were accessed and not the entire population of agile practitioners in South Africa. The researcher chose LinkedIn as it is a professional networking social platform. The sample size could be increased in future research to reach a large audience which could make generalization easier. The data collection instrument of the questionnaire proved to be very useful as every question was answered by every respondent, however including a qualitative data collection technique will provide more depth to the findings. Open-ended questions will enable future researchers to retrieve reasons for the selection of the critical success factors as the respondents and will give greater insight into their perceptions as they will be able to elaborate. It will allow future researchers to attain an understanding of the different scenarios that exist in agile projects

Future research could undertake the limitation of this study by adopting a mixed-method approach via a qualitative survey which will enable the respondents to provide their insight and experience in projects. If qualitative methods were introduced via a mixed-methods study, together with quantitative questions, this would allow the individual responses to be unbiased as open-ended questions allow for varying responses and possible new findings.

It was also noted in a study by Tam, Moura, Oliveira, and Varajão (2018) which investigated the single dimension of people factors that contribute to the success of agile projects. The findings of the study showed that managers should strive to choose a competent team and to encourage collaboration with the customer. Tam et al. (2018) also suggested that future research should include other people's characteristics example the individual's ability to react to change which is important for agile success. This corresponds with the finding of this study as the people factor was perceived as the most important critical success factor for agile success and is indicative of future research that could consider additional people attributes.

The following is noted:

- A larger sample size could be sourced in future research to enable the results to be more generalizable by extending the survey to other social platforms.
- Future studies could extend the research to expand on the test automation and cloud computing factors as these are new phenomena in small scale agile projects. Expanding this topic via qualitative methods, for example, face-face interviews the true perception or feelings of the respondents will surface by open-ended questions that will provide detailed feedback. More information is required on these two factors and the closed-ended questions were unable to provide further insight into the actual practice of these two factors in organisations. It is possible that organisations are still being accustomed to cloud computing and test automation and they are still

5.5 CONCLUSION

The findings of this study reveal that there is a gap between the perceived importance and perceived performance of the critical success factors that affect agile projects. The results of this study serve as an indication to organisational stakeholders on the critical success factors, the most important CSFs, the performance of these factors, and the gap between the importance and performance factors. The gap that exists between the importance and performance factors will motivate organisations to close this gap. It will allow for greater emphasis to be placed on the performance of critical success factors in agile projects which could result in a higher success rate. This means that organisations can focus attention on the performance of the most important critical success factors which will result in successful agile projects. The failure of agile projects could be the result of the performance of critical success factors that are being overlooked in agile projects. The hierarchy of people, technical, and test automation factors should be a benchmark for organisations to use as the most important factors that should have the highest support from the organisation and project team members to ensure agile success. This study included two additional constructs which were test automation and cloud computing. Test automation was ranked as a critical factor on one hand, while cloud computing was found to be less critical to agile success. The findings also showed the gap that exists between the importance and performance of the critical success factors. To minimize this gap, organisations should emphasize the importance of the critical success factors that are integral to agile project success and ensure that their resources are invested in the performance of these factors.

REFERENCES

Abd El Hameed, T., Abd El Latif, M., & Kholief, S. (2016). Identify and Classify Critical Success Factor of Agile Software Development Methodology Using Mind Map. *International Journal of Advanced Computer Science and Applications*, *7*(5), 83–92. https://doi.org/10.14569/ijacsa.2016.070513

Ambler, S. (2006). The Agile Unified Process (AUP) Home Page. The Agile Unified Process (AUP), 1–4. Retrieved May, 28, from http://www.ambysoft.com/unifiedprocess/agileUP.html

Abdalhamid, S., & Mishra, A. (2017). Factors in Agile methods adoption. *TEM Journal*, *6*(2), 416–421. https://doi.org/10.18421/TEM62-29

About The Standish Group - The Standish Group. (2019). Retrieved November 20, 2020, from https://www.standishgroup.com/about

Abrahamson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). Agile software development methods: Review and analysis. *VTT Publications*, 112. http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf

Ahmad, G., Rahim Soomro, T., & Mehmood Raza Naqvi, S. (2016). An Overview: Merits of Agile Project Management Over Traditional Project Management in Software Development. *Journal of Information and Communication Technology*, *10*(1), 105–120. http://ibt.edu.pk/qec/jict/10.1/10.pdf

Ahmad, S., Wasim, S., Irfan, S., Gogoi, S., Srivastava, A., & Farheen, Z. (2019). Qualitative v/s. Quantitative Research- A Summarized Review. *Journal of Evidence Based Medicine and Healthcare*, *6*(43), 2828–2832. https://doi.org/10.18410/jebmh/2019/587

Aldahmash, A., Gravell, A. M., & Howard, Y. (2017). A review on the critical success factors of agile software development. Communications in Computer and Information Science, 748(August 2017), 504–512. https://doi.org/10.1007/978-3-319-64218-5_41

Alfaki, M.A.I., Ali, O., Babiker, A.E., & Ibrahim, A.O. (2016). Agile RACI Model for Extreme Programming Method. International Journal of Advanced Research in Compter Science and Software Engineering, 6(5), 60–64.

Alhir, S. (2005). The Agile Unified Process (AUP). Retrieved June 2, 2021, from https://www.methodsandtools.com/archive/archive.php?id=21

Alizamini, F.G., Pedram, M.M., Alishahi, M. & Badie, K. (2010). Data quality improvement using fuzzy association rules, pp. V1-468-V1- 472

Almeida, F. (2017). Challenges in Migration from Waterfall to Agile Environments. *World Journal of Computer Application and Technology*, *5*(3), 39–49. https://doi.org/10.13189/wjcat.2017.050302

Almudarra, F., & Qureshi, B. (2015). Issues in adopting agile development principles for mobile cloud computing applications. Procedia Computer Science, 52(1), 1133–1140.

https://doi.org/10.1016/j.procs.2015.05.131

Altman, D.G., & Bland, J.M. (1995). Statistics notes: the normal distribution. BMJ (Clinical research ed.), 310(6975), 298. https://doi.org/10.1136/bmj.310.6975.298

Apuke, O. D. (2017). Quantitative Research Methods : A Synopsis Approach. *Kuwait Chapter of Arabian Journal of Business and Management Review*, *6*(11), 40–47. https://doi.org/10.12816/0040336

Arikpo, I. I., & Osofisan, A.O. (2019). Introducing Agile Software Development into an Organisation : The Role of the Customer Parts of Customer Relationship to be affected. *Computing & Information Systems*, *14*(2), 28–38.

Awad, M. A. (2005). A Comparison between Agile and Traditional Software Development Methodologies. The University of Western Australia, 1(June), 1–300.

Bacon-Shone, J. (2015). Introduction to Quantitative Research Methods pres. In *Loughborough University* (Issue February). https://doi.org/10.13140/2.1.4466.3040

Bartlett, J.E., Kotrlik, J.W. and Higgins, C.C. (2001) Organisational Research: Determining Appropriate Sample Size in Survey Research. Information Technology, Learning, and Performance Journal, 19, 43-50.

Beck, K. (1999). "Embracing Change With Extreme Programming." IEEE Computer 32(10): 70- 77.

Bhardwaj, M. (2014). An Introduction to Agile Methodology. Retrieved April 21, 2020, from https://manasbhardwaj.net/introduction-agile-methodology/

Biemer, P. P., & Lyberg, L. (2003). Introduction to survey quality. Hoboken, NJ: Wiley. https://doi.org/10.1002/0471458740

Boehm, B. (1988). A spiral model of software development and enhancement. *Computer*, *21*, 61-72.

Boehm, B. (2002). Get Ready for Agile Methods, with Care. *Computer, 35*, 64-69.

Bullen, P.B. (2014). How to choose a sample size (for the statistically challenged). Retrieved June 28, 2020, from http://www.tools4dev.org/resources/how-to-choose-a-sample-size/

Buresh, D. L. (2008). Customer Satisfaction and Agile Methods. *Reliability Society 2008 Annual Technology Report*, 1–8. https://pdfs.semanticscholar.org/5347/cf49af4dd491ba591c0453a488288e3a0baa.pdf

Caralli, R.A. (2004). The Critical Success Factor Method : Establishing a Foundation for Enterprise Security Management. Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Institute. July.

Cervone, H. F. (2011). Understanding agile project management methods using Scrum. *OCLC Systems and Services*, *27*(1), 18–22. https://doi.org/10.1108/10650751111106528

Ceschi, M., Sillitti, A., Succi, G., & De Panfilis, S. (2005). Project management in plan-based and agile companies. IEEE software, vol. 22(3), pp.21-27, 2005

Chiyangwa, T. B. (2017). *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 UNIVERS*. *January*.

Chiyangwa, T. B., & Mnkandla, E. (2017). Modelling the critical success factors of agile software development projects in South Africa. *SA Journal of Information Management*, *19*(1), 1–8. https://doi.org/10.4102/sajim.v19i1.838

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. https://doi.org/10.1016/j.jss.2007.08.020

Cohn, M. (2002). Mountain Goat Software. Retrieved August 10, 2019, from http://www.mountaingoatsoftware.com/

Collins, E. F., & De Lucena, V. F. (2012). Software test automation practices in agile development environment: An industry experience report. *2012 7th International Workshop on Automation of Software Test, AST 2012 - Proceedings*, *December*, 57–63.

Creswell, J. (2002). Educational research: Planning, conducting, and evaluating Quantitative and Qualitative research. Upper Saddle River, NJ: Merrill Prentice Hall.

Creswell, J. W. (2009). Research Design: Qualitative, Quantitative, and Mixed Methods Approaches (3rd ed.). Thousand Oaks, CA: Sage Publications.

Cronbach's Alpha in SPSS statistics - procedure, output and interpretation of the output using a relevant example | Laerd statistics. (2018). SPSS Statistics Tutorials and Statistical Guides | Laerd Statistics. Retrieved November 8, 2020, from https://statistics.laerd.com/spss-tutorials/cronbachs-alpha-using-spss-statistics.php

Cruth, M. (2020). The Spotify model | Atlassian. Atlassian. Retrieved September 10, 2020, from https://www.atlassian.com/agile/agile-at-scale/spotify

Demirkan, H., & Spohrer, J. (2015). T-Shaped Innovators: Identifying the Right Talent to Support Service Innovation, Research-Technology Management, 58:5, 12-15. https://doi.org/10.5437/08956308X5805007

Department of Linguistics - Home | Department of Linguistics. (n.d.). (2008). https://Www.Ling.Upenn.Edu/~clight/Chisquared.Htm. Retrieved September 17, 2019, from https://www.ling.upenn.edu/

Devi, V. (2013). Traditional and Agile Methods: An Interpretation. Retrieved March 12, 2020, from http://www.scrumalliance.org/community/articles/2013/january/traditional-and-agile-methods-an-interpretation

Doody, O., & Doody, C. M. (2015). Conducting a pilot study: Case study of a novice researcher. *British Journal of Nursing*, *24*(21), 1074–1078. https://doi.org/10.12968/bjon.2015.24.21.1074

Dustin, E., Rashka, J., & Paul, J. (2008). Automated Software Testing: Introduction, Management, and Performance. https://www.researchgate.net/publication/31705330

Dyba, T., & Dingsoyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, *50*(9–10), 833–859. https://doi.org/10.1016/j.infsof.2008.01.006

Easterby-Smith, M., Thorpe, R., Jackson, P. and Lowe, A. (2012) Management Research (4th edn). London: S

El-Gilany, A-H. (2018). Sample size & sampling methods. https://www.researchgate.net/publication/323457758

Etikan, I., Musa, S. A., & Alkassim, R. S. (2017). *Comparison of Convenience Sampling and Purposive Sampling Comparison of Convenience Sampling and Purposive Sampling*. *January 2016*. https://doi.org/10.11648/j.ajtas.20160501.11

Explorable.com. 2019. *ANOVA - Statistical Test - The Analysis Of Variance*. Retrieved March 12, 2020, from https://explorable.com/anova

Franken, S., Kolvenbach, S., Prinz, W., Alvertis, I., & Koussouris, S. (2015). CloudTeams: Bridging the Gap between Developers and Customers during Software Development Processes. *Procedia Computer Science*, *68*(December), 188–195. https://doi.org/10.1016/j.procs.2015.09.234

Garousi, V., Tarhan, A., Pfahl, D., Coşkunçay, A., & Demirörs, O. (2018). Correlation of critical success factors with success of software projects: an empirical investigation. *Software Quality Journal*, *27*(1), 429–493. https://doi.org/10.1007/s11219-018-9419-5

Gates, L. P. (2010). Strategic Planning with Critical Success Factors and Future Scenarios : An Integrated Strategic Planning Framework. November.

Griethuijsen, R.A.L.F., Eijck, M. W., Haste, H., Brok, P. J., Skinner, N. C., Mansour, N., Gencer, A. S., & BouJaoude, S. (2015). Global patterns in students' views of science and interest in science. Research in Science Education, 45(4), 581–603. https://doi.org/10.1007/s11165-014-9438-6

Grubb, P., & Takanga, A. A. (2003). *Software maintenance: Concepts and practice*. River Edge, N.J: World Scientific.

Gündüz, M., Nielsen, Y., & Özdemir, M. (2013). Quantification of Delay Factors Using the Relative Importance Index Method for Construction Projects in Turkey. *Journal of Management in Engineering*, *29*(2), 133–139. https://doi.org/10.1061/(asce)me.1943-5479.0000129

Hair, J.F., Black, W.C., Babin, B.J. and Anderson, R.E. (2010) Multivariate Data Analysis. 7th Edition, Pearson, New York.

Hakansson, A. (2013). Portal of Research Methods and Methodologies for Research Projects and Degree Projects. *Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering FECS'13*, 67–73. http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-136960

Harisha, K.R. (2018). Software Development Process and Its Importance. Retrieved October 5, 2020, from https://www.streetdirectory.com/travel_guide/135488/software/software_development_process_and_its_importance.html

Hatkar, K.B., & Hedaoo, K.N. (2016). Hedaoo N A Delay Analysis By Using Relative Importance Index Method in. *International Journal of Civil Engineering and Concrete Structure*.

Hossen, M.M., Kang, S. & Kim, J. 2015. Construction schedule delay risk assessment by using combined AHP- RII methodology for an international NPP project. Nuclear Engineering and Technology 47(3): 362–379. http://dx.doi.org/10.1016/j.net.2014.12.019

Islam, M. (2016). *A Systematic Literature Review on the Critical Factors that Contribute to Success of Agile Development Projects*. *June*, 40. http://hdl.handle.net/2077/46944

Jalali, S., & Wohlin, C. (2010). Agile practices in global software engineering - A systematic map. *Proceedings - 5th International Conference on Global Software Engineering, ICGSE 2010*, *June*, 45–54. https://doi.org/10.1109/ICGSE.2010.14

Javanmard, M., & Alian, M. (2015). Comparison between Agile and Traditional software development methodologies. Cumhuriyet University Faculty of Science Science Journal. 36.

Johnson, B., & Christensen, L. (2008). Educational research: Quantitative, qualitative, and mixed approaches (p. 34). Thousand Oaks, CA: Sage Publications.

Joseph, N., Marnewick, C., & Santana, M. J. (2016). Agile software development and it project performance in South Africa: A positive relationship? *IAMOT 2016 - 25th International Association for Management of Technology Conference, Proceedings: Technology - Future Thinking*, *August 2019*, 338–358. on at: https://www.researchgate.net/publication/318206568

Kabir, S.M.S. (2016). Basic Guidelines for Research an Introductory Approach for all. *Basic Guidelines for Research: An Introductory Approach for All Disciplines*, *August*, 557.https://www.researchgate.net/publication/325846997_METHODS_OF_DATA_COLLECTION

Kalem, S., Donko, D., & Boskovic, D. (2013). Agile methods for cloud computing. *2013 36th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2013 - Proceedings*, *April*, 1079–1083.

Karhu, K., Repo, T., Taipale, O., & Smolander, K. (2009). Empirical observations on software testing automation. *Proceedings - 2nd International Conference on Software Testing, Verification, and Validation, ICST 2009*, *May*, 201–209. https://doi.org/10.1109/ICST.2009.16

Kaur, N., & Singh, G. (2016). Critical Success Factors in Agile Software Development Projects : A Review. International Journal on Emerging Technologies, 7(1), 1–4.

Kenton, W. (2020). T-Test Retrieved September 3, 2019, from .https://www.investopedia.com/terms/t/t-test.asp

Khan, S., & Jamal, H. (2014). "Advantages of Using prototype software development model instead of waterfall model? - Engineering Questions Answers QnA - Agricultural, Electrical, Civil, Computer, Mechanical", Enggpedia.com, 2014. Retrieved June 2, 2021, from http://www.enggpedia.com/answers/2057/advantages-prototype- software-development-instead-waterfall. [Accessed: 2- June- 2021]

Kropp, M., & Meier, A. (2015). Agile Success Factors. *Clinical and Experimental Rheumatology*, *32 Suppl 8*(2), 1–4. http://www.ncbi.nlm.nih.gov/pubmed/24743859

Kulathunga, D., & Ratiyala, S. D. (2018). Key Success Factors of Scrum Software Development Methodology in Sri Lanka. *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, *45*(January), 234–252.

Kumar, D., & Mishra, K. K. (2016). The Impacts of Test Automation on Software's Cost, Quality and Time to Market. *Procedia Computer Science*, *79*, 8–15. https://doi.org/10.1016/j.procs.2016.03.003

Lam, S.L., Cheung, R., Wong, S., & Chan, E.S. (2013). A survey study of critical success factors in information system project management.

Leau, Y., Loo, W.K., Tham, W.Y., & Tan, S.F. (2012). Software Development Life Cycle AGILE vs Traditional Approaches. 37(Icint), 162–167.

Lee, D. K., In, J., & Lee, S. (2015). Standard deviation and standard error of the mean. *Korean Journal of Anesthesiology*, *68*(3), 220–223. https://doi.org/10.4097/kjae.2015.68.3.220

Lichtman, M. (2006). Qualitative research in education: A user's guide (pp.). Thousand

Lindvall, M., Basili, V., Boehm, B., Costa, P., Dangle, K., Shull, F., Tesoriero, R., Williams, L. & Zelkowitz, M. (2002). Empirical Findings in Agile Methods. Proceedings of Extreme Programming and Agile Methods - XP/Agile Universe 2002, pp. 197-207

Linke, K. (2019). Traditional and Agile Management Approaches. *12th ILERA European Congress, At Heinrich Heine University (HHU) Düsseldorf, Germany*, *September*, 15.

Lippert, M., Wolf, H., & Roock, S. (2002). Extreme programming in action: Practical experiences from real world projects: John Wiley & Sons, Inc.

Liu, Q. (2012). A Survey Data Quality Strategy. IR Application. https://www.researchgate.net/publication/278156884_A_Survey_Data_Quality_Strategy

Maarit, L. (2014). Characteristics and Principles of Scaled Agile. Lecture Notes in Business Information Processing. 199. https://doi.org/10.1007/978-3-319-14358-3_2.

Madadipouya, K. (2015). A review on the strategic use of IT applications in achieving and sustaining competitive advantage. International Journal of Managing Public Sector Information and Communication Technologies, 6(2), 21–30. https://doi.org/10.5121/ijmpict.2015.6202

Martin, P. & Bateson, P. (1986). *Measuring Behaviour: An Introductory Guide*. Cambridge: Cambridge University Press. https://www.researchgate.net/publication/278156884

Mbelli, T. M., & Jaintylal Hira, J. (2016). *The Perceptions of Agile Methodology in South Africa*. 219–227. https://doi.org/10.5121/csit.2016.60119

Memon, M. A., Ting, H., Cheah, J.-H., Thurasamy, R., Chuah, F., & Cham, T. H. (2020). Sample Size for Survey Research: Review and Recommendations. *Journal of Applied Structural Equation Modeling*, *4*(2), i–xx. https://doi.org/10.47263/jasem.4(2)01

Mersino, A. 2018. Agile Project Success Rates are 2X Higher than Traditional Projects. Retrieved June 20, 2020, from https://vitalitychicago.com/blog/agile-projects-are-more-successful-traditional-projects/

Miller, G. (2013). *Agile problems, challenges, & failures Agile Problems, Challenges, & Failures*. https://www.researchgate.net/publication/335475075

Misra, S. C. (2007). Adopting agile software development practices: success factors, changes required, and challenges. In *ProQuest Dissertations and Theses*. http://eserv.uum.edu.my/docview/304888117?accountid=42599%5Cnhttp://dl.acm.org/citation.cfm?id=1329710

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. https://doi.org/https://doi.org/10.1016/j.jss.2009.05.052

Mkrtchyan, R. (2020). The Code and Fix Model. Retrieved March 10, 2020, from https://productcoalition.com/the-code-and-fix-model-2cabd4c48166

Mohammed, A. M., & Abushama, H.M. (2013). *Popular Agile Approaches in Software Development : Review and Analysis. August 2013*. https://doi.org/10.1109/ICCEEE.2013.6633925

Mujumdar, A., Masiwal, G., & Chawan, P. M. (2012). Analysis of various Software Process Models. http://www.researchgate.net/profile/Pramila_Chawan/publication/267427007_Analysis_of_various_Software_Process_Models/links/54f0aa150cf2f9e34efd0776.pdf

Munassar, N. M. A., & Govardhan, A. (2010). A comparison between five models of software engineering. IJCSI, 5, 95-101.

Nasir, M. H. N., & Sahibuddin, S. (2011). Critical success factors for software projects: A comparative study. Scientific Research and Essays, 6(10), 2174–2186.

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, *48*(5), 72–78. https://doi.org/10.1145/1060710.1060712

Noruwana, N., & Tanner, M. (2012). Understanding the structured processes followed by organisations prior to engaging in agile processes: A South African Perspective. *South African Computer Journal*, *48*(1). https://doi.org/10.18489/sacj.v48i1.74

Oleksandrova, O. (2018). Infographic: A Brief History of Software Development Methodologies. Retrieved May 21, 2021, from https://intetics.com/blog/a-brief-history-of-software-development-methodologies

Paquet, J. (2019). SAFe vs Spotify. Retrieved June 10, 2021, from https://www.myagilepartner.com/blog/index.php/2019/05/31/safe-vs-spotify/

Paulk, M. C. (1997). Software Process Proverbs. *Crosstalk - The Journal of Defense Software Engineering*, *10*(1), 1–6. https://www.researchgate.net/publication/260402731

Pfeffer, J. (1998). The human equation: building profits by putting people first. Boston, MA, Harvard Business School Press, cop.

Popli, R., Anita, & Chauhan, N. (2013). A Mapping Model for Transforming Traditional Software Development Methods to Agile Methodology. *International Journal of Software Engineering & Applications*, *4*(4), 53–64. https://doi.org/10.5121/ijsea.2013.4405

Pressman, R.S. (2005). Software engineering: A practitioner's approach (6th ed.), New York, McGraw-Hill.

Rather, M. A., & Bhatnagar, V. (2016). *A comparative study of sdlc model*. *October 2015*.

Rockart, J. (1979). Chief executive's define their own data needs. Harvard Business Review, 57(2), 81-93.

Roscoe, J. T. (1975). Fundamental research statistics for the behavioral sciences (Second ed.). New York: Holt Rinehart and Winston.

Saunders, M., Lewis, P., & Thornhill, A. (2019). Chapter 4: Understanding research philosophy and approaches to theory development. In *Research Methods for Business Students* (Issue March). https://www.researchgate.net/publication/330760964

Sayeed, A., Hassan, N., & Muttoo, M. (2017). International Journal of Computer Science and Mobile Computing Agile Methodology Utilizing Cloud Computing. *International Journal of Computer Science and Mobile Computing*, *6*(6), 307–310. www.ijcsmc.com

Sedgwick, P. (2013). Convenience sampling. *Bmj*, *347*(oct25 2), f6304–f6304. https://doi.org/10.1136/bmj.f6304

Siegle, 2020. T-Test. Retrieved August 10, 2020, from https://researchbasics.education.uconn.edu/t-test/

Singh, S. (2018). Statistics: Descriptive and Inferential. Retrieved August 3, 2020, from https://towardsdatascience.com/statistics-descriptive-and-inferential-63661eb13bb5

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. *Journal of Systems and Software*, *86*(6), 1663–1678. https://doi.org/10.1016/j.jss.2013.02.027

Steinberg, D.H., & Plamer, D.W. (2003). Extreme software engineering a hands-on approach: Prentice-Hall, Inc.

Stoica, M., Mircea, M., & Ghilic-Micu, B. (2013). Software Development: Agile vs. Traditional. Informatica Economica, 17(4/2013), 64–76. https://doi.org/10.12948/issn14531305/17.4.2013.06

Sudhakar, G. P. (2012). A model of critical success factors for software projects. Journal of Enterprise Information Management, 25(6), 537–558. https://doi.org/10.1108/17410391211272829

Sullivan, G.M. and Artino Jr, A.R. (2013). Analyzing and Interpreting Data From Likert-Type Scales. https://www.jgme.org/doi/full/10.4300/JGME-5-4-18

Taber, K.S. (2013). Classroom-based research and evidence-based practice: an introduction (2nd ed.). London: Sage.

Taber, K.S. (2018). The Use of Cronbach's Alpha When Developing and Reporting Research Instruments in Science Education. *Research in Science Education*, *48*(6), 1273–1296. https://doi.org/10.1007/s11165-016-9602-2

Tam, C., Moura, E. J. da C., Oliveira, T., & Varajão, J. (2020). The factors influencing the success of on-going agile software development projects. *International Journal of Project Management*, *38*(3), 165–176. https://doi.org/10.1016/j.ijproman.2020.02.001

Tanner, M., & Von Willingh, U. (2014). Factors Leading To the Success and Failure of Agile Projects Implemented in Traditionally Waterfall Environments. *Human Capital without Border: Knowledge and Learning for Quality of Life. Management, Knowledge and Learning International Conference 2014, January 2014,* 693–701. https://www.researchgate.net/publication/312187721

Taylor, P. C., & Medina, M. N. D. (2013). Educational research paradigms: from positivism to multiparadigmatic. *Journal of Meaning-Centered Education*, *1*(2007), 1–16.

Thummadi, B. V., Shiv, O., Berente, N., & Lyytinen, K. (2011). *Service-Oriented Perspectives in Design Science Research*. *6629*(June 2014). https://doi.org/10.1007/978-3-642-20633-7

Totten, J. (2017). Critical Success Factors for Agile Project Management in Non-Software Related Product Development Teams. *Dissertation*, *1*, 1–152

Tumbas, P., & Matković, P. (2006). Agile vs Traditional Methodologies in Developing Information Systems. *Management Information Systems*, *1*, 15–24.

Turetken, O., Stojanov, I., & Trienekens, J. J. M. (2016). Assessing the adoption level of scaled agile development: a maturity model for Scaled Agile Framework. *Journal of Software: Evolution and Process*, *29*(6). https://doi.org/10.1002/smr.1796

Uludag, O., Kleehaus, M., Xu, X., & Matthes, F. (2017). Investigating the Role of Architects in Scaling Agile Frameworks. *Proceedings - 2017 IEEE 21st International Enterprise Distributed Object Computing Conference, EDOC 2017*, *2017-Janua*(March 2019), 123–132. https://doi.org/10.1109/EDOC.2017.25

Vithana, V.N., Fernando, G.S.S., & Kapurubandara, M. (2015). Success Factors for Agile Software Development A Case Study from Sri Lanka. *International Journal of Computer Applications*, *113*(17), 10–18. https://doi.org/10.5120/19917-2056

Wan, J., & Wang, R. (2010). Empirical Research on Critical Success Factors of Agile Software Process Improvement. *Journal of Software Engineering and Applications*, *03*(12), 1131–1140. https://doi.org/10.4236/jsea.2010.312132

What is incremental model advantages, disadvantages and when to use it. (2008). Retrieved June, 1, 2021, from http://tryqa.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/)

Yaghoobi, T. (2018). Prioritizing key success factors of software projects using fuzzy AHP. Journal of Software: Evolution and Process, 30(1), 1–11. https://doi.org/10.1002/smr.1891

Younas, M., Jawawi, D. N. A., Ghani, I., Fries, T., & Kazmi, R. (2018). Agile development in the cloud computing environment: A systematic review. *Information and Software Technology*, *103*(September), 142–158. https://doi.org/10.1016/j.infsof.2018.06.014

Yu, J. (2018). Research Process on Software Development Model. IOP Conference Series: Materials Science and Engineering, 394(3). https://doi.org/10.1088/1757-899X/394/3/032045

Zhang, Z., & Sharifi, H. (2000). A methodology for achieving agility in manufacturing organisations. International Journal of Operations and Production Management, 20(4), 496–513. https://doi.org/10.1108/01443570010314818

APPENDICES

APPENDIX A: INFORMED CONSENT

UKZN HUMANITIES AND SOCIAL SCIENCES RESEARCH ETHICS

COMMITTEE (HSSREC)

APPLICATION FOR ETHICS APPROVAL

For research with human participants

Information Sheet and Consent to Participate in Research

Date:

Greetings,

My name is Yeshmeeta Peters, (Student No: 983170355), a Master's student from the discipline of Information Systems and Technology, School of Management, IT and Governance of the University of KwaZulu-Natal. My contacts details for myself and the academic discipline at UKZN are listed below:

Student email: 983170355@stu.ukzn.ac.za

Student Mobile Number: +27846484463

Discipline of Information Systems and Technology: +27312607051

You are being invited to consider participating in a study that involves the study of the critical success factors that affect agile software development.

The objective of the study is to contribute by leveraging the knowledge from current Agile software practitioners to propose a critical success model that will assist in achieving Agile project success. The title of the project is:

Critical success factors for agile success and the performance of these factors in agile software development organisations

The current aspect of the study is directed at obtaining insight into your experience of the critical success factors that affect agile projects. This insight will be collected using a questionnaire comprising of closed-ended questions. The duration of your participation, if you choose to participate in the study, is expected to be about 15-20 minutes. Participation in the study is optional and you are free to withdraw at any time.

We envisage that the information that you provide will be valuable in creating a hierarchy of importance for the critical success factors that affect agile projects. It is also noted that the outcome of this study will provide a valuable contribution to the general study of agile success.

This study has been ethically reviewed and approved by the UKZN Humanities and Social Research Ethics Committee (approval number: HSSREC/00001942/2020).

In the event of any problems or concerns/questions you may contact the researcher on; Cell: 0846484463 or E-mail: 983170355@stu.ukzn.ac.za or the UKZN Humanities & Social Sciences Research Ethics Committee, contact details as follows:

Humanities and Social Science Ethics (HSSREC)

Research Office,

Govan Mbeki Building, Westville Campus

Private Bag X54001, DURBAN 4000

Tel: 031 260 8350

Email: HSSREC@ukzn.ac.za

Your participation in the study is voluntary and by participating, you are granting the researcher permission to use your responses. You may refuse to participate or withdraw from the study at any time with no negative consequence. There will be no monetary gain from participating in the study. Your anonymity will be maintained by the researcher and the School of Management, I.T. & Governance and your responses will not be used for any purposes outside of this study.

All data, both electronic and hard copy, will be securely stored during the study and archived for 5 years. After this time, all data will be destroyed.

If you have any questions or concerns about participating in the study, please contact me at the number listed above.

Sincerely

Y. Peters

APPENDIX B: CONSENT TO PARTICIPATE

I …………………………………… have been informed about the study entitled, Critical success factors for agile success and the performance of these factors in agile software development organisations

I understand the purpose and procedures of the study. I have been given an opportunity to ask questions about the study and have had answers to my satisfaction.

I declare that my participation in this study is entirely voluntary and that I may withdraw at any time without affecting any of the benefits that I usually am entitled to.

If I have any further questions/concerns or queries related to the study I understand that I may contact the researcher on Cell: 0846484463 or E-mail: 983170355@stu.ukzn.ac.za

If I have any questions or concerns about my rights as a study participant, or if I am concerned about an aspect of the study or the researchers then I may contact:

Humanities and Social Science Ethics (HSSREC)

Research Office,

Govan Mbeki Building, Westville Campus

Private Bag X54001, DURBAN 4000

Tel: 031 260 8350

Email: HSSREC@ukzn.ac.za

____________________ ____________________

Signature of Participant **Date**

____________________ ____________________

Signature of Witness **Date**

(Where applicable)

____________________ ____________________

Signature of Translator **Date**

(Where applicable)

APPENDIX C: RESEARCH INSTRUMENT (QUESTIONNAIRE)

DEMOGRAPHIC INFORMATION

Industry Type of Organisation	
Information Services	
Banking	
Legal	
Advertising	
Hospitality	
Manufacturing	
Retail	
Distribution	
Municipal/Government	
NGO	
Consulting services	
Building/Construction	
Health Care	
Other	
Age Group	
18-20	
21-25	
26-30	
31-35	
36-40	
41-45	
Above 45	
Number of employees in the organisation	
< than 10	

10-20	
21 -4 0	
41 -1 0 0	
101-500	
501 -1000	
Greater than 1000	
Number of team members in projects	
5 or fewer team members	
5-10 team members	
10-15 team members	
More than 15 team members	
Your role in the organisation/project	
Organisation management	
Functional Manager	
Project Administrator	
Project Manager	
Scrum Master	
Team Lead	
Developer/Tester	
Business Analyst	
Data Analyst	
Client	
Other (Please specify)	
Number of years' experience in agile software development projects	
0-2	
3-5	
6-8	
9-11	
12-14	
Above 14	
Number of agile projects that you were involved in	
0-5	
6-10	
11-14	
15-20	
21-25	
Above 25	

Type of agile development	
Extreme Programming	
Scrum	
Dynamic Systems Development Method	
Feature-Driven Development	
Adaptive Software Development	
Lean software development	
Crystal	
Gender	
Male	
Female	
Level of tertiary education	
None	
Certificate	
Diploma	
Degree	
Other	
Organisation location	
Gauteng	
Western Cape	
Mpumalanga	
Eastern Cape	
Northern Cape	
Free State	
KwaZulu Natal	
Limpopo	

Please evaluate the factors according to your assessment of the success factors in agile software development in column 1 in terms of importance & please evaluate the reality of these factors in your organisation in column 2 in terms of performance/implementation											
	Column 1: Importance						**Column 2: Implementation**				
	Strongly disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**		**None**	**Poor**	**Fair**	**Good**	**Very Good**
1. Organisational structure	**1**	**2**	**3**	**4**	**5**	**1. Organisational structure**	**1**	**2**	**3**	**4**	**5**
1. Strong executive support is important for agile project success						1. How well was strong executive support provided during agile projects?					
2. Project manager or project sponsor support is important for agile success						2. How strong was the support of the project manager or sponsor?					
3. Organisational culture supporting teamwork and collaboration over the rank of leadership is important for agile success						3. How well did the organisation support teamwork and collaboration over the rank in leadership?					
4. Culture that supports face-to-face communication contributes to agile success						4. How well did the organisation support face-to-face communication?					
5. Agile development support throughout the organisation is important for agile success						5. How well was agile development supported throughout the organisation?					
6. Team that is collocated and share the working space						6. How well did the organisation support collocated teams					

influences agile success						and shared working spaces?					
7. Office with the appropriate environment is important for agile development						7. How well did the organisation support the appropriate environment for agile development?					
8. A reward system in line with agile methodologies is important for agile success						8. How strong was the organisation's support towards a reward system in agile development?					
9. Organisation support for the decision making of developers is important for agile success						9. How well did the organisation support the decisions of developers?					
10. Organisation support for rapid delivery is important for agile success						10. How strongly did the organisation support rapid delivery?					
2. People						**2. People**					
11. Highly skilled and competent team members are important for agile success						11. How well were the team members skilled in agile methods?					
12. Team members who are strongly motivated contribute to agile success						12. How strongly were the team members motivated?					
13. Managers skilled in agile processes influence the agile success						13. How skilled were managers in agile processes?					
14. Managers who have an adaptive or light-touch style of management contribute to agile success						14. How strong was the need for managers to have an adaptive or light-touch way of management?					
15. Teams that self-organize						15. How well did team					

and manage themselves are important for agile success						members self-organize and manage themselves?					
16. Good relationship with the customer is important for agile success						16. How good was the relationship with the customer?					
17. Team members with strong interpersonal skills are important for agile success						17. How strong were the team members' interpersonal skills?					
3. Process - tools/methods used in projects						**3. Process - tools/methods used in projects**					
18. Organisation that follows an agile-oriented requirement management process is important for agile success						18. How well did the organisation follow an agile-oriented requirement management process?					
19. Organisation that follows an agile-oriented project management process is important for agile success						19. How well did the organisation follow an agile-oriented project management process?					
20. Organisation that follows agile-oriented configuration management process is important for agile success						20. How well did the organisation follows an agile-oriented configuration management process?					
21. Projects that have daily stand up meetings & face to face feedback sessions supports the agile success						21. How strong was the support for daily stand up meetings & face to face feedback sessions to support agile?					
22. Organisation that has a daily working roster with no overtime						22. How well did the organisation support a normal working					

supports the agile success						schedule with no overtime?					
23. Customer who is strongly committed & present within the projects is important for agile success						23. How strong was customer commitment and presence in projects?					
4. Technical Factors						**4. Technical Factors**					
24. Coding standards that are properly defined at the outset of the project contributes to agile success						24. How well did the organisation support coding standards to be properly defined at the start of the project?					
25. Projects with the goal of simple design influences agile success						25. How well did the organisation support a simple design process?					
26. Project with rapid refactoring processes is important for agile success						26. How well were rapid refactoring processes supported?					
27. Documentation that is only compiled as necessary influences agile success						27. How strong was the support for only necessary documentation?					
28. Software that is delivered regularly is important for agile success						28. How strong was the support to deliver software regularly?					
29. Projects that deliver the most important functionality first influence the agile success						29. How strongly was the support to deliver the most important functionality first?					
30. Accurate integration testing is important for agile success						30. How strong was the support for accurate integration testing?					

31. Team provided with adequate technical training influences agile success						31. How well were the teams provided with adequate training?						
32. Projects adopting development standards that are in line with agile practices of delivering in quick increments contributes to agile success						32. How well did projects support the adoption of agile development standards and practices of delivering in quick increments?						
5. Project						**5. Project**						
33. Projects that are not life-critical succeed with agile methods						33. How well did non-life critical projects succeed with agile?						
34. Projects that allow for changing requirements succeed using agile methods						34. How well did projects that allow for changing requirements succeed with agile?						
35. Projects that can adapt to a change in schedule succeed in agile						35. How well did projects that can adapt to changing schedules succeed in agile?						
36. Small project teams succeed in agile						36. How well were small project teams succeed in agile?						
37. Projects that do not have many separate teams succeed in agile						37. How well were fewer teams supported?						
38. Projects cost evaluated at the start of the project is important for agile success						38. How strong was the support for project cost to be evaluated at the start of the project?						

39. Risk analysis conducted at the start of the project is important for agile success						39. How strong was the support for risk analysis to be conducted at the start of the project?					
6. Test Automation						**6. Test Automation**					
40. Increasing test coverage by using automated testing is important for agile success						40. How well was automated testing implemented to increasing testing coverage in agile projects					
41. Test-driven development by using automated testing is important for agile success						41. How well was automated testing implemented to increasing test-driven development in agile projects					
42. Continuous integration testing by using automated testing is important for agile success automated acceptance testing						42. How well was automated testing implemented for continuous integration testing in agile projects					
43. Acceptance testing using automated testing is important for agile success						43. How well was automated testing implemented for acceptance testing in agile projects					
44. Regression testing using automated testing is important for agile success						44. How well was automated testing implemented for regression testing in agile projects					
7. Cloud Computing						**7. Cloud Computing**					
45. Infrastructure as a Service provided by cloud computing is important for agile success						45. How well was Infrastructure as Code implemented in agile projects?					
46. Testing using cloud computing is important for agile success						46. How well was cloud computing testing implemented in agile projects.					
47. Team collaboration using cloud computing is important for agile success						47. How well was cloud computing team collaboration implemented in agile projects?					

48. Automation of the end to end process (build, test, run) using cloud computing is important for agile success						48. How well was cloud computing end to end automation implemented in agile projects?					
49. User tool integration using cloud computing is important for agile success						49. How well was cloud computing user tool integration implemented in agile projects?					

The project was successful in:	**Strongly disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
1. Quality (delivering a good product or project outcome)					
2. Scope (meeting all requirements and objectives)					
3. Time (delivering on time)					
4. Cost (delivering within estimated cost and effort)					

APPENDIX D: ETHICAL APPROVAL LETTERS

UNIVERSITY OF KWAZULU-NATAL
INYUVESI YAKWAZULU-NATALI

01 October 2020

Mrs Yeshmeeta Deodutt Peters (983170355)
School Of Man Info Tech &Gov
Westville Campus

Dear Mrs Peters,

Protocol reference number: HSSREC/00001942/2020
Project title: Critical success factors and the performance of these factors in agile software development projects
Degree: Masters

Approval Notification – Expedited Application

This letter serves to notify you that your application received on 14 September 2020 in connection with the above, was reviewed by the Humanities and Social Sciences Research Ethics Committee (HSSREC) and the protocol has been granted **FULL APPROVAL** on the following condition:

Any alteration/s to the approved research protocol i.e. Questionnaire/Interview Schedule, Informed Consent Form, Title of the Project, Location of the Study, Research Approach and Methods must be reviewed and approved through the amendment/modification prior to its implementation. In case you have further queries, please quote the above reference number. PLEASE NOTE: Research data should be securely stored in the discipline/department for a period of 5 years.

This approval is valid until 01 October 2021.
To ensure uninterrupted approval of this study beyond the approval expiry date, a progress report must be submitted to the Research Office on the appropriate form 2 - 3 months before the expiry date. A close-out report to be submitted when study is finished.

All research conducted during the COVID-19 period must adhere to the national and UKZN guidelines.

HSSREC is registered with the South African National Research Ethics Council (REC-040414-040).

Yours sincerely,

[redacted]

Professor Urmilla Bob
(University Dean of Research)

/dd

Humanities and Social Sciences Research Ethics Committee
Postal Address: Private Bag X54001, Durban, 4000, South Africa
Telephone: +27 (0)31 260 8350/4557/3587 **Email:** hssrec@ukzn.ac.za **Website:** http://research.ukzn.ac.za/Research-Ethics

Founding Campuses: Edgewood Howard College Medical School Pietermaritzburg Westville

15 September 2021

Mrs Yeshmeeta Deodutt Peters (983170355)
School Of Man Info Tech & Gov
Westville Campus

Dear Mrs Peters,

Protocol reference number: HSSREC/00001942/2020
Project title: Critical success factors and the performance of these factors in agile software development projects
Amended title: Critical success factors that influence the performance of agile software development methodologies in organisations

Approval Notification – Amendment Application

This letter serves to notify you that your application and request for an amendment received on 01 September 2021 has now been approved as follows:

- Change in title

Any alterations to the approved research protocol i.e. Questionnaire/Interview Schedule, Informed Consent Form; Title of the Project, Location of the Study must be reviewed and approved through an amendment /modification prior to its implementation. In case you have further queries, please quote the above reference number.

PLEASE NOTE: **Research data should be securely stored in the discipline/department for a period of 5 years.**

All research conducted during the COVID-19 period must adhere to the national and UKZN guidelines.

Best wishes for the successful completion of your research protocol.

Yours faithfully

[redacted]

Professor Dipane Hlalele (Chair)

/dd

Humanities & Social Sciences Research Ethics Committee
UKZN Research Ethics Office Westville Campus, Govan Mbeki Building
Postal Address: Private Bag X54001, Durban 4000
Tel: +27 31 260 8350 / 4557 / 3587
Website: http://research.ukzn.ac.za/Research-Ethics/