Identification and prioritisation of success factors in agile software development in the South African software development industry

by

Ashley Gwangwadza
u16206186

Submitted in partial fulfilment of the requirements for the degree

MIT in Information Systems

in the

FACULTY OF ENGINEERING, BUILT ENVIRONMENT AND INFORMATION TECHNOLOGY

at the

UNIVERSITY OF PRETORIA

Study leader:

Hanslo, Ridewaan

Date of submission
31 August 2023

Declaration regarding Plagiarism

The Department of Informatics emphasises integrity and ethical behaviour with regard to the preparation of all written assignments.

Although the lecturer will provide you with information regarding reference techniques, as well as ways to avoid plagiarism, you also have a responsibility to fulfil in this regard. Should you at any time feel unsure about the requirements, you must consult the lecturer concerned before submitting an assignment.

You are guilty of plagiarism when you extract information from a book, article, web page or any other information source without acknowledging the source and pretend that it is your own work. This does not only apply to cases where you quote verbatim, but also when you present someone else's work in a somewhat amended (paraphrased)x format or when you use someone else's arguments or ideas without the necessary acknowledgement. You are also guilty of plagiarism if you copy and paste information directly from an electronic source (e.g., a web site, e-mail message, electronic journal article, or CD ROM), even if you acknowledge the source.

You are not allowed to submit another student's previous work as your own. You are furthermore not allowed to let anyone copy or use your work with the intention of presenting it as his/her own.

Students who are guilty of plagiarism will forfeit all credits for the work concerned. In addition, the matter will be referred to the Committee for Discipline (Students) for a ruling. Plagiarism is considered a serious violation of the University's regulations and may lead to your suspension from the University. The University's policy regarding plagiarism is available on the Internet at *http://upetd.up.ac.za/authors/create/plagiarism/students.htm*.

I (full name & surname):	Ashley Shumirayi Gwangwadza
Student number:	U16206186

Declare the following:

1. I understand what plagiarism entails and am aware of the University's policy in this regard.
2. I declare that this assignment is my own original work where someone else's work was used (whether from a printed source, the Internet or any other source); due acknowledgement was given and reference was made according to departmental requirements.
3. I did not copy and paste any information directly from an electronic source (e.g., a web page, electronic journal article or CD ROM) into this document.
4. I did not make use of another student's previous work and submitted it as my own.
5. I did not allow and will not allow anyone to copy my work with the intention of presenting it as his/her own work.

A.S Gwangwadza — Signature

2023/08/31 — Date

TABLE OF CONTENTS

Commented [RH1]: Note it is typical to have the following sections:
- Introduction
- Literature review
- Theoretical framework (maybe if large enough, else can be part of lit review)
- Research methodology
- Research results
- Discussion (discussion of findings, which is discussing the results)
- Conclusion

Commented [AG2]: Results then discussion of findings

APPENDICES

LIST OF FIGURES

Commented [RH3]: Take the names out of the figure titles in the list of figures, apply throughout….

LIST OF TABLES

LIST OF ABBREVIATIONS

ASD	Agile Software Development
CSF	Critical Success Factors
FDD	Feature-Driven Development
IS	Information Systems
IT	Information Technology
LSD	Lean Software Development
MVP	Minimum Viable Product
SDLC	Software development life cycle
SLR	Systematic Literature Review
TAM	Technology Acceptance Model
TPB	Theory of Planned Behaviour
TDD	Test Driven Development
TRA	Theory of Reasoned Action
UAT	User Acceptance Testing
UI	User Interface
UTAUT	Unified Theory of Acceptance and Use of Technology
UX	User Experience
XP	Extreme Programming

IDENTIFICATION AND PRIORITISATION OF SUCCESS FACTORS IN AGILE SOFTWARE DEVELOPMENT IN THE SOUTH AFRICAN SOFTWARE DEVELOPMENT INDUSTRY

ABSTRACT

Agile software development is a methodology initiated to offer "agility" by addressing the rigidity of traditional methodologies to limit the associated negative effects. It has been increasingly adopted in the South African software development industry. However, there is no comprehensive research on the factors that affect the success of agile software development projects in a South African context and the priority of these factors. This study addresses this gap through a case study of a South African software development organisation. The study identifies and prioritises the critical success factors of agile software development in the South African software development industry.

Fifteen semi-structured interviews were conducted with agile practitioners in various roles in the case study organisation to identify the critical success factors. Twenty-five critical success factors were grouped into six categories: organisational, team, customer, process, technical and project. Based on the findings, the research proposes a framework for critical success factors in agile software development in South African software development organisations. The factors were then ranked based on the combined frequency of mentions of the factor and its descriptive attributes. Organisational culture was the highest-ranked factor, showing that the Agile project perceived it to have the most influence on Agile project success. The ranking shows that the organisational and team factors have the most highly ranked factors, while the project category has the least ranked factors. The research findings provide information that, if used by agile practitioners in South Africa, may increase the success of the Agile software development project in the South African software development industry.

Keywords: Agile methodologies, software development projects, critical success factors, Scrum, case study.

1 INTRODUCTION

Software is vital in all areas of the modern world; however, software development is far from being deemed a perfect process (Arcos-Medina and Mauricio, 2020). The software development process still suffers from age-old challenges of cost overruns, project delays, and unfulfilled user requirements (Barki et al., 1993). Success in software development projects is challenging to attain (Tam et al., 2020). This is demonstrated by the numerous software projects that are abandoned, delayed, or rejected (Arcos-Medina and Mauricio, 2020). According to Johnson (2018), 64% of software projects were challenged or failed since they were not on time, on target (scope) and within budget (cost).

Organisations spend a lot of money implementing Agile development processes, hoping the software projects will be finished on time at a lesser cost (Ambler, 2009). The Agile methodology is incorporated into software development to create business value, deliver working software regularly and improve quality (Shakya and Shakya, 2020).

Agile methods have been increasingly adopted as the software development methodology of choice; however, some researchers criticise it because there are as many success stories as failure stories (Russo, 2021). Johnson (2018) found that only 48% of Agile projects were successful, and the remaining 62% were dismissed as failed or challenged.

Therefore, how to achieve success in Agile software development needs to be further explored. Several quantitative studies have identified the factors contributing to global Agile project success (Chiyangwa, 2017, Chow and Cao, 2008, Nasehi, 2013, Stankovic et al., 2013, Tam et al., 2020: 2020). The quantitative method explains results in figures and heavily depends on statistics. However, a qualitative method helps to provide a deep understanding of the explanations for certain phenomena and scenarios (Asnawi, 2012).

Kaplan and Maxwell (2005) argue that when textual data is quantified, the goal of comprehending an event from the participants' perspective and its specific social and institutional environment is essentially lost. Due to this, there has been a general shift in Information systems research from quantitative technology issues to qualitative organisational issues and understanding the context of IS (Myers and Avison, 2002). Qualitative approaches help explain users' and practitioners' behaviours in relation to the system, the system's success and failures, and even what is regarded as success (Kaplan and Shaw, 2004). Thus, this study will employ a qualitative approach due to the nature of the research topic with the primary goal of obtaining a comprehensive understanding of the critical success factors in Agile software development in their environment.

Commented [AG4]: Check citation

1.1 BACKGROUND

The Agile methodology implies the ability to survive and be successful in an environment of continuous change (Chow and Cao, 2008). The implication stems from "Agile", which means to be responsive and flexible. The Agile methodology is beyond just a set of methods and procedures; it is a specific attitude, a way of thinking coined the Agile mindset (Miler and Gaida, 2019). Incorporating the Agile method with system development is expected to aid organisations in becoming more productive in a rapidly changing and competitive business environment (Nasehi, 2013). Agile methods are sets of new approaches in software development that have numerous practices such as Scrum, Extreme Programming (Akbar et al.), Dynamic System Development (DSDM), Lean Development (LD), Feature Driven Development (FDD) and Crystal (Asnawi, 2012).

Agile software development is a software development methodology that was initiated to promote collaboration between users and developers, bring short development cycles, and respond to instabilities in a dynamic environment (Arcos-Medina and Mauricio, 2020). It aims to provide "agility" by addressing the inflexibility innate to traditional methodologies and curbing the consequent negative impact (Tam et al., 2020). Agile software development is built upon an Agile philosophy detailed in the Agile Manifesto (Dikert et al., 2016).

The Agile Manifesto was first published in 2001 by a group of IT professionals. The manifesto details principles and value statements to guide software development (Syeda, 2018). The Agile principles are listed in Appendix L. Since the manifesto's release, Agile has begun to gain momentum in the software development industry (Aldahmash et al., 2017). It motivates software development to focus on producing business value and providing software to end-users within short intervals (Shameem et al., 2017).

The critical success factors theory was presented to find the vital areas an organisation needs to focus on to achieve a specific goal (Shameem et al., 2017). Critical success factors are the few crucial areas where positive outcomes are vital for a manager to achieve his goals, according to Bullen and Rockart (1981). Chow and Cao (2008) were among the first researchers to identify critical success factors in Agile software development based on a survey of 109 Agile practitioners.

Since then, many researchers have explored the different facets of critical success factors. Riaz et al. (2018) identified the social success factors based on communication in Agile software development projects by providing an insight into how the factors contribute to project success. Shehzad and Kausar (2021) present a subset of critical success factors related to migration challenges from traditional to Agile methodologies, such as management commitment, organisational culture, training, communication and collaboration, change management and mindset. Aldahmash (2018) reviewed the critical success factors of Agile software development and found that there is a need to explore how the factors relate to each other regarding ranking and inter-relationships to maximise the benefits of Agile software development.

1.2 PROBLEM STATEMENT

Software performs basic or sophisticated tasks everywhere nowadays (Bogopa and Marnewick, 2022). However, there are still high rates of software development projects in South Africa and globally (Javani et al., 2022). Research shows that there is hope as Agile software projects display better success rates than traditional methodologies.

The findings by Khoza and Marnewick (2020) showed that Agile projects showed a success rate of 12% more than Waterfall projects. Mkoba and Marnewick (2020) state that 71% of organizations globally use agile approaches to increase IT project success rates, and of the available agile approaches, 75% of South African organizations use scrum.

Researchers have gained considerable interest in Agile software development methodologies in the last few years (Akhtar et al., 2022). The increased popularity is also seen in researchers exploring the factors that contribute to success in Agile software development projects (Chiyangwa, 2017, Chow and Cao, 2008, Ghayyur et al., 2018, Hamdani and Butt, 2017, Kelle et al., 2015, Misra et al., 2006, Qatanani et al., 2021).

Although several quantitative studies have been conducted to identify the critical success factors, some partially contradictory findings emerge from this quantitative research, indicating the need for further research (Hummel and Epp, 2015). For example, Tam et al. (2020) state that some of their findings contradict the findings of Misra et al. (2009). Both were survey-based quantitative studies, but Tam et al. (2020) did not find personal characteristics and societal culture to be critical success factors, although they were identified in Misra et al. (2009) study. Stankovic et al. Stankovic et al. (2013) did not confirm that all the factors identified by Chow and Cao (2008) can be considered critical success factors in the Yugoslavian IT industry.

The inconsistencies in the quantitative research findings imply a need for the critical success factors of Agile software development to be analysed using different methods and in the context of different countries. Nguyen (2016) suggested that future research focuses on Identifying critical success factors in other countries. Research in this area is lacking in a South African Context. According to Khoza and Marnewick (2020), IT projects in South Africa organisations have the following failure rates: small organisations at 50.2%, medium-sized organisations at 64.2%, large organisations at 16.9%, and the overall failure rate for South African organisations is 53.7%. Thus, there is a clear need for improvement.

Peters (2020) states that there is insufficient research into the priority of the critical success factors affecting Agile projects. This is also believed to be true for South Africa. The research aims to provide software practitioners with information that can increase the probability of success in future Agile software development projects (Tam et al., 2020). This will be done as explanatory research to explain which factors In agile software development are critical for success and which are deemed more important.

Thus, assessing the critical success factors of Agile software development in South Africa may provide insights to help Agile practitioners in South Africa complete the development process more successfully. Consequently, once the factors are identified, this study aims to rank these success factors concerning the level of importance for the success of Agile projects.

Unlike most previous predominately quantitative studies, these tasks will be undertaken using a qualitative method. A qualitative study will assist in resolving previous contradictory findings by allowing the researcher to study the phenomenon and its context in depth (Myers, 2019).

1.1 PURPOSE OF THE STUDY

This study aimed to identify and prioritise the critical success factors of Agile software development in the South African software development industry. Although several aspects influence Agile success, some factors are vital to project success. It is poised that the hierarchy of the critical success factors will assist practitioners in focusing on the most significant factors.

1.2 RESEARCH OBJECTIVES

Main Research Objective:
This research aims to identify and establish a hierarchy of the critical success factors that affect Agile software development success as perceived by Agile practitioners in the South African software development industry.

The following specific research objectives were set based on the research questions:

a) To identify the critical factors in the success of Agile software development recognised by Agile practitioners in the South African software development industry.
b) To determine the priority of the identified critical success factors.

1.3 RESEARCH QUESTIONS

The research questions this study aims to address are the following:

Main Research Question:
What are the critical success factors and their priority in Agile software development in the South African software development industry?

The following research sub-questions are investigated to aid in answering the research question:

a) What are the critical factors in the success of Agile software development recognised by Agile practitioners in the South African software development industry?
b) What is the priority of the identified critical success factors?

1.4 ASSUMPTIONS

The case study method is based on several assumptions. Goddard and Melville (2004) list some of these assumptions, and the following apply to this study:

i. The assumption of human consistency, irrespective of the fact that human behaviour can vary depending on circumstances. The assumption of studying the natural history of the social unit.
ii. The assumption of a comprehensive examination of a unit. In this case, the organisation chosen is the unit.

Commented [AG5]: Check in-text referencing

Commented [AG6]: Make this statement make sense

1.5 LIMITATIONS

Qualitative findings are highly contextualised and case-dependent (Patton, 1999). This case study was conducted in a South African organisation and cannot be fully generalised globally. Thus, the limitation of a case study is that since it involves one organisation's behaviour, the findings' generality is unclear (Heeager and Nielsen, 2013). A case study may suggest what might be found in comparable organisations, but further research would be required to validate whether the outcomes from one study generalise in a different place (Simon and Goes, 2013). However, the researcher believes much can be learned from a single case study. A single case study allows for an exhaustive study of the chosen organisations, allowing the researcher to fully understand the behaviour pattern of the concerned organisation and the phenomena of interest (Goddard and Melville, 2004). Additionally, it provides in-depth analysis and rich insights (Darke et al., 1998)

1.6 CONTRIBUTION

This research will contribute to the scientific knowledge of Agile methodologies, particularly the critical success factors 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 their priorities. To potentially reduce the failure of Agile projects, Agile teams must focus on the key factors contributing to project success (Peters, 2020).

1.7 BRIEF CHAPTER OVERVIEW

Chapter 1 – Introduction

The research study's first chapter introduces the research topic and gives background information. The research objectives and questions are also presented in this chapter. It provides an overview of the critical points in the study and the research outline.

Chapter 2 – Literature Review:

This chapter is the literature review, which critically evaluates the current literature related to the topic. It discusses software development and the software development life cycle, Agile methodologies and their impact on software development, and a comparison of Agile and traditional software development. A systematic literature review establishes the basis for Agile project success and the crucial success factors essential to the study's topic.

Chapter 3 - Methodology:

The Research Methodology chapter describes the research methods employed in this study and provides reasoning for the methodology chosen. It also outlines the ethical considerations for the research.

Chapter 4 –Results:

This chapter presents the data collected through semi-structured interviews at a software development organisation in South Africa. This section presents an analysis of the case study organisation that participated in the case study. The section presents the descriptive statistics of the interviewee demographics. Additionally, it describes how the collected data was consolidated to produce the critical success factors and presents the synthesised and prioritised lists of critical success factors.

Chapter 5 – Discussion of Findings:

This chapter discusses the results of the thematic analysis and answers the research question. It presents the framework developed from the data analysis and discusses the identified CSFs in their respective categories. This chapter also discusses the priority of the identified critical success factors.

Chapter 6 – Conclusion:

The concluding chapter of this study presents a summary of the findings, the research contribution, limitations of the research, recommendations for future research and concluding remarks.

2 LITERATURE REVIEW

2.1 INTRODUCTION

This chapter reviews the existing literature on the critical success factors of Agile software development (ASD). A Literature review was done on research areas related to the research topic. Secondly, a systematic literature review was conducted to identify the critical success factors in Agile software development. This structured approach was taken by similar studies investigating critical success factors; thus, it is poised to be an appropriate approach. The chapter begins with background information on software development. The second section provides background information on the systematic literature review process and how it was conducted. The third section discusses the findings of the systematic literature review. The fourth section discusses the research gap identified in the literature. The fifth section presents the theoretical framework, and the final section concludes the chapter.

2.2 BACKGROUND ON SOFTWARE DEVELOPMENT

This section discusses the current research on areas related to the research topic in the following areas: software development in general, Agile methodologies, Agile software development and critical success factors, each with its subsection.

2.2.1 Software development

In the modern age, organisations and people cannot function without software development (Peters, 2020). Organisations have come to rely on software for office work, administration, banking, and other areas because computers and software are integral elements of business and technology (Munassar and Govardhan, 2010). People are becoming more dependent on technology, with 75% of a person's life connected to technology (Dwivedi et al., 2022). This reliance increases the need for software to be developed to satisfy people's needs (Peters, 2020). In turn, the number of organisations producing software solutions have increased (Munassar and Govardhan, 2010).

Nevertheless, the software development process remains complex (da Silva and dos Santos, 2015). According to Tam et al. (2020), completing software development projects is challenging.

Chow and Cao (2008) emphasise that realising how software development can be improved to prevent failure is one of the biggest hurdles software development organisations face. Thus, this research aims to potentially suggest ways of improving software development through the identified critical success factors. Some researchers argue that a systematic development approach that emphasises understanding the scope and complexity of the development process is necessary for project success (Leau et al., 2012). One method employed in software development is the "Software Development Life Cycle" (SDLC). The following section discusses the software development lifecycle and traditional software development methods.

2.2.2 Software Development Life Cycle

The "Software Development Life Cycle" (SDLC) refers to the activities that outline the phases involved in building software (Asnawi, 2012). The SDLC aims to deliver high-quality work in a timely and cost-effective manner while exceeding customer expectations (Dwivedi et al., 2022). The SDLC includes the essential activities to implement software development successfully (Yu, 2018), as shown in Figure 1.

Figure 2.1: Software Development Lifecycle

Source: (Stoica et al., 2013:65)

Dwivedi et al. (2022) state that the SDLC enhances the general development process and software quality. The author argues that the SDLC ensures that all functionalities, user needs, goals, and outcomes are achieved. There are various methods for the software development lifecycle, each of which was developed with specific goals in mind (Stoica et al., 2013). Each model adheres to a set of steps tailored to a particular type of project to enable the success of system development (Dwivedi et al., 2022). Most system developers currently use either traditional or Agile development as their SDLC approaches (Leau et al., 2012). These approaches are detailed in the following sections.

2.2.3 Traditional Software Development Methodologies

According to Boehm (2002), traditional software development methodologies (TSDMs) are meticulously planned, codified processes, detailed, documented and designed in-depth at the beginning of a project. The foundation of traditional methods is a set of sequential steps (Leau et al., 2012). This traditional approach starts the project with a clear definition of user requirements, and clients specify their expectations immediately (Gaborov et al., 2021). The prevalent criticism is that the traditional approach does not consider the probability that requirements might change (Peters, 2020). These traditional methodologies include the spiral, unified process, and waterfall models.

Spiral Model. In an effort to blend the benefits of top-down and bottom-up principles, it integrates components of both design and prototyping-in-stages (Awad, 2005). This model combines the Waterfall and unified models with a primary focus on risk management (Shaikh and Abro, 2019).

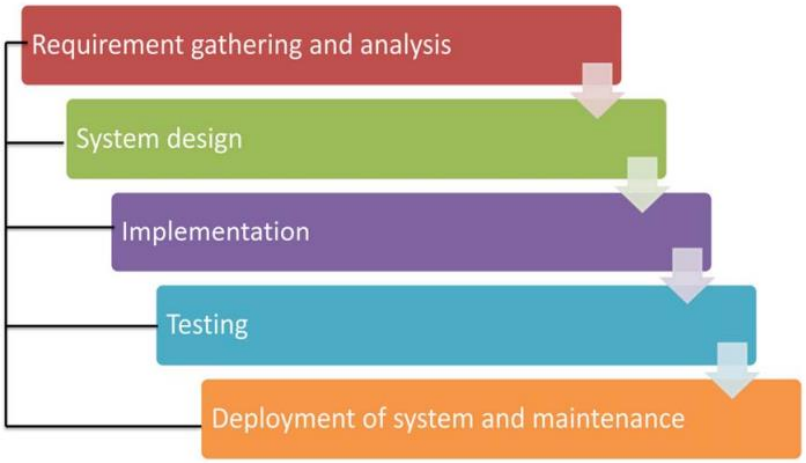
Unified Process Model. It is a well-clear model that outlines what must be done, when it must be done, and who will execute it in a project (Shaikh and Abro, 2019). The Unified Process model organises all activities, including modelling, into workflows and carries them out incrementally and iteratively (Awad, 2005).

Waterfall Model. According to research, Waterfall is the most widely utilised traditional method in software development (Olorunshola and Ogwueleka, 2022). It is often compared with scrum research to depict the differences between traditional and agile methods. Thus, it will be discussed below in detail, given its significance.

2.2.4 Waterfall Model

Winston W. Royce defined the waterfall model in 1970 (Stoica et al., 2013). According to Thummadi et al. (2011), the waterfall approach is a one-way, top-down, non-iterative method for creating software. Figure 2 illustrates the various non-overlapping stages of the pure waterfall lifecycle.

Figure 2.2: Phases of the waterfall model



Source: (Stoica et al., 2013:66)

As shown in the diagram, each phase must be finished before the start of another (Olorunshola and Ogwueleka, 2022). Implementing change within the development lifecycle can be challenging when such an approach is taken because the project's success depends on understanding all requirements before development begins (Leau et al., 2012).

One of the advantages of the waterfall method is that it makes it easier to estimate project expenses, establish a timeline, and allocate resources appropriately (Leau et al., 2012). However, foreseeing every aspect and requirement of a project at the start is challenging since more information is learned as the project progresses (Yu, 2018). It is exceedingly challenging to keep up with the continuously changing requirements of the business world (Asnawi, 2012). In an effort to address the dilemma of traditional approaches that do not allow for enough flexibility in the contemporary corporate environment, Agile methods arose in the mid-1990s (Marnewick and Labuschagne, 2009).

2.3 AGILE METHODOLOGIES

Over the past ten years, Agile methods have been employed extensively in software engineering (Lukusa et al., 2020), making them the dominant methodology used in software development projects (Alami et al., 2022). They indicate a significant shift from the traditional software development approach toward a more engineering approach (Mbelli and Hira, 2016). Although the concept of agility was not new, it gained widespread acceptance when it was first presented in 2001 as the Agile Manifesto (Shehzad and Kausar, 2021). It catalysed the Agile movement in the software industry (Aldahmash, 2018).

Examples of methods that fall under agile include Feature Driven Development (FDD), Crystal, Lean Software Development (LSD), Dynamic System Development (DSDM), Extreme Programming (XP), and Scrum (Asnawi, 2012)

Feature-Driven Development. It is an agile technique for developing object-oriented software (Hanslo et al., 2019). Agile development and model-driven design are combined. The initial object model, feature-based segmentation, and iterative design are highlighted (Joseph and Santana, 2016). It is said to be the most appropriate for critical systems.

Extreme Programming (XP). It was first made known by (Beck et al., 2001), and it has now become a commonly used Agile software development technique. The following are some of the primary XP practices according to (Hanslo et al., 2019):

- At one location, five to ten programmers work alongside a client representative.
- Development occurs over several iterations that may or may not be released, producing incremental functionality.
- User stories are used to specify requirements, each of which contains a section of new functionality required by the user.
- Programmers collaborate in teams, adhere to stringent coding guidelines, and conduct unit tests.
- Clients take part in acceptance testing.
- Throughout the project, requirements, architecture, and design are revealed.

Crystal Methods. Clear, yellow, orange, red, and blue are crystal family techniques created for various team sizes and criticality (Joseph and Santana, 2016). The term 'crystal' denotes the heaviness of the technique (Asnawi, 2012). The approach focuses on the importance of the team and its members' experiences (Asnawi, 2012). For small-scale teams creating non-critical software, Crystal Clear is the most suitable agile method (Joseph and Santana, 2016).

Lean software development (LDS). It embraced the concepts and methods of lean manufacturing, whose principal goal is to remove waste and anything that does not hinder development (Asnawi, 2012). Lean manufacturing principles are modified for software development (Joseph and Santana, 2016).

Dynamic System Development Method (DSDM). With an incremental and iterative procedure, DSDM is an expansion of the Rapid Application Development (RAD) framework (Asnawi, 2012). Following what is known as the "80% rule," DSDM only creates the work necessary for each increment to move on to the next increment (Hanslo et al., 2019). Thus, 80% of the system is built in 20% of the time.

Scrum. It is the most well-known agile software development approach worldwide (Peters, 2020). According to the research by Joseph et al. (2016)
Scrum is also the most used agile methodology in South Africa, with 82.9% usage. Additionally, it is the main methodology used in the organisation selected for the case study. Thus, because of its significance to the study, Scrum is discussed in detail in the section below.

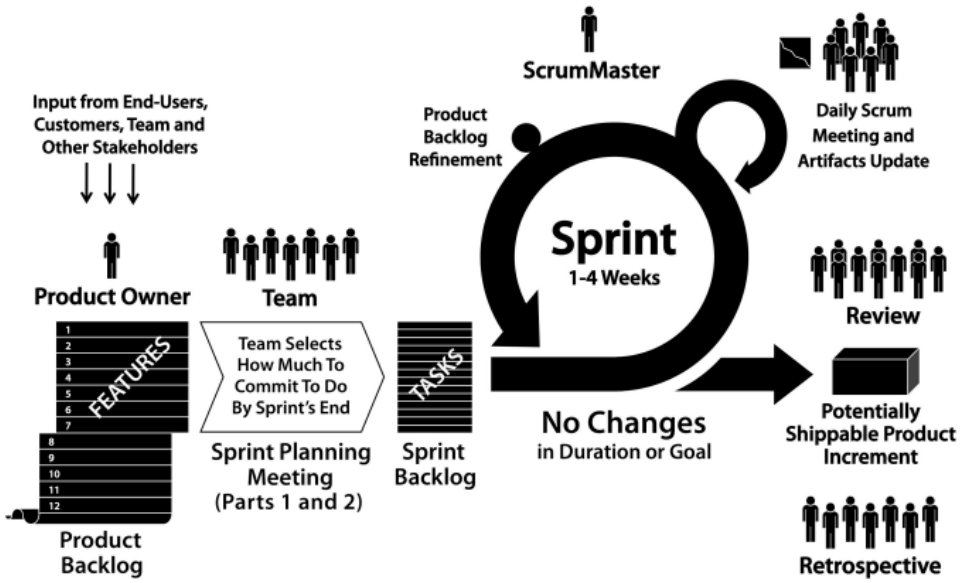
2.3.1 Scrum

Scrum is an iterative, incremental process (Gaborov et al., 2021) developed to address the rapidly evolving business requirements (Asnawi, 2012). Scrum aims to help development teams focus on set goals and reduce time spent on unimportant tasks (Stoica et al., 2013). The method does not specify the steps involved in the development process; it only offers a set of managerial recommendations and is one

of the most popular Agile methods (Gaborov et al., 2021). It focuses on project management, including mechanisms for "empiric process control," where feedback loops are the key component for circumstances when initial planning is challenging (Stoica et al., 2013). Scrum breaks the project's stages into a series of iterations called "sprints," where software development starts when the product backlog is established (Asnawi, 2012).

The Scrum process specifies four meetings and three roles. The Scrum meetings outlined are Daily Scrum, Sprint Planning, Scrum Retrospective and Sprint Review (Linke, 2019). The roles include Scrum Master, Product Owner and Development team (Peters, 2020). The key events and roles in Scrum are summarised in Figure 3.

Figure 2.3: Scrum Summary



Source: (Deemer et al., 2010:348)

2.3.2 The Agile Manifesto

The Agile Manifesto outlines the values and twelve principles that constitute the basis of the Agile movement (Alami et al., 2022). According to Fowler and Highsmith (Fowler and Highsmith, 2001), the purpose of the Agile Manifesto is "to uncover better ways of developing software by doing it and helping others do it". The values detailed in the Agile manifesto are shown in Table 1.

Table 2.1: The Values of The Agile Methodology

They value:	Over
Individuals and interaction	processes and tools
Working software	comprehensive documentation
Customer collaboration	contract negotiation
Responding to change	following a plan

Source: (Beck et al., 2001:1)

The manifesto also detailed the principles promoted by the 17 specialists based on best practices and their prior successes and failures with numerous software development projects regarding what functions and what does not (Misra et al., 2006). Below is a summary of the principles detailed in the Agile Manifesto according to Beck et al. (2001):

- Customer satisfaction is the highest priority.
- Welcoming changes in requirements, even when they occur late in the development process.
- Regular delivery of functional software
- Developers and business individuals should work collectively throughout the project.
- A supportive environment with motivated people
- Face-to-face interaction is the most efficient form of communication.
- Working software serves as a measure of progress.
- Sustainable development
- Attention to good design and technical excellence.

- Simplicity is essential.
- Self-organising teams
- Team behaviour should be adjusted to be more effective.

This Agile Manifesto is the foundational idea for Agile methodologies (Shameem et al., 2020).

2.4 AGILE SOFTWARE DEVELOPMENT

Agile software development is a lightweight methodology developed to get around the drawbacks of traditional development approaches, cut costs and overheads, and provide flexibility to accommodate requirement changes at any point (Al-Saqqa et al., 2020). It aims to promote user and developer collaboration, take advantage of quick development cycles, and adapt to changes in a dynamic environment (Arcos-Medina and Mauricio, 2020).

Agile Software Development (ASD) offers an iterative approach to producing effective and efficient software (Ghayyur et al., 2018).In an Agile environment, the software development process is team-oriented (Kamal et al., 2020). Instead of being a concern for process modelling, Agile software development must be viewed as a cultural topic of project teams and software-producing enterprises and how these choreographs collaborate (Kuhrmann et al., 2021). Agile software development has significantly impacted how software is built worldwide (Dyba and Dingsoyr, 2009). This section discusses the difference between traditional and Agile software development and the impact of Agile on software development.

2.4.1 Agile Software Development Versus Traditional Software Development

Most system developers currently use either traditional or Agile development as their SDLC approaches (Leau et al., 2012). Khoza and Marnewick (2020) explain that the fundamental assumptions of traditional methods are that systems are entirely predictable, specifiable, and developed after careful and comprehensive planning. In contrast, Agile methods assume that high-quality adaptive software development is

grounded on principles of ongoing design refinement and testing based on rapid feedback (Khoza and Marnewick, 2020).

According to Akbar et al. (2020), the main difference between Agile and traditional software development is that Agile yields less documentation, completes projects quicker, allows for requirement changes, increases customer satisfaction, improves product quality, and provides customers with transparency. Other differences between traditional and agile software development are found in scientific literature and are summarised in Table 2.

Table 2.2: Comparison of Agile and Traditional Software Development

Parameter	Agile	Traditional	Source
Approach	Adaptive Approach	Predictive approach	(Stoica et al., 2013).
Planning Scale	Short Term	Long Term	(Khoza and Marnewick, 2020)
Management Style	Leadership, Collaboration	Command, Controlling	(Al-Saqqa et al., 2020)
User Requirements	Acquired Iteratively	Complete user requirements defined before development	(Leau et al., 2012)
Customers	Not empowered and with minimal commitment	Empowered and dedicated	(Khoza and Marnewick, 2020)
User Involvement	At the start of the project	Throughout the project	(Varghese et al., 2022)
Ownership	Shared ownership. Each team member is accountable for their contribution.	lies in the project manager	(Tam et al., 2020)
Team Size	Small	Medium	(Al-Saqqa et al., 2020)
Project Size	Small or Medium	Large	(Al-Saqqa et al., 2020)
Documentation	Just enough essential specifications and documentation	Extensive documentation and specifications	(Radhakrishnan et al., 2021)
Development Orientation	Customer Oriented	Process Oriented	(Al-Saqqa et al., 2020)

Parameter	Agile	Traditional	Source
Retrospectives	Done after each meeting in a sprint retrospective meeting	Lessons learned session at the end of the project	(Overhage and Schlauderer, 2012)
Testing	Development and testing are done concurrently	Testing is done after the development	(Akbar et al., 2020)
Product delivery	Incremental delivery	One-time delivery	(Tam et al., 2020)
Success probability	High	Very High	(Varghese et al., 2022)

According to (Dwivedi et al., 2022), Agile methodologies are selected when:

- General requirements and objectives are known.
- The project will likely require flexibility and changes throughout since they can save costs, speed up production, and ensure on-time delivery.

While traditional methods are selected when:

- It is clear precisely what needs to be done at each project stage.
- The project cannot be broken up into smaller components.
- The project customer is only concerned with the finished output.
- Team communication is informal.

In the scientific literature, Scrum is the most used Agile methodology, and Waterfall is the most used traditional methodology in software development (Gaborov et al., 2021). Agile approaches are more flexible and, thus, generally better than traditional ones because they ensure the timely delivery of high-quality software within budget and increase the chances of project success (Gaborov et al., 2021).

2.4.2 The Impact of Agile Software Development on Project Success

There is no denying that the 20 years of Agile software development have brought about many improvements (Kuhrmann et al., 2021). Altameem (2015) states that Agile software development concepts foster more trust and commitment, enabling people to carry out their responsibilities successfully and create sustainable development.

Because of the trust, the development team can collaborate directly with clients to meet their needs and create a system that gives them a competitive advantage.

The Agile methodology's primary goal is to encourage the swift and expeditious development of products that address customers' actual demands. (Hamdani and Butt, 2017). It is widely acknowledged that Agile methods, as opposed to conventional ones, are better equipped to manage uncertain needs and produce high-quality software quickly and within budget (Tam et al., 2020). This makes Agile methodologies popular because they lower development costs and manage modifications necessary at later stages of the project (Peters, 2020).

Agile methods are today used to boost productivity and improve development quality. (Salikhov et al., 2020). Teams can develop software rapidly and react to changes that may occur during a project by following the values and principles proposed by Agile techniques and practices (Arcos-Medina and Mauricio, 2020). Whether using Scrum, Extreme Programming, or other Agile approaches, adaptability is an essential characteristic that can be noted in Agile software development methodologies (Malik et al., 2019). This adaptability increases the probability of project success (Gaborov et al., 2021)

Many improvements in software development can be attributed to Agile methodologies (Hamdani and Butt, 2017). According to the most recent findings, software projects using Agile principles typically succeed more often than projects using the Waterfall method (Khoza and Marnewick, 2020). Meenakshi et al. (2020) compared the success rates of traditional and agile software development methods. The findings are summarised in Table 3.

Table 2.3: Comparison of success of traditional and Agile methods

Project Size	Traditional Method Success	Agile Method Success
Small	44%	58%
Medium	7%	27%
Large	3%	18%

Source: (Meenakshi et al., 2020:34)

Evidently, using Agile methodologies increases the likelihood of project success. However, the success rates shown are still not impressive and indicate a need to understand how to improve project success even when using Agile methodologies. The consensus is that Agile techniques outperform traditional methodologies in many ways, but some challenges still hinder project success (Stoica et al., 2013).

According to a study by Mbelli and Hira (2016), Agile techniques were shown to have a 50% acceptance rate in South Africa. The acceptance rate can indicate that Agile practitioners in South Africa may not have confidence in utilising Agile methodologies. This confidence can be boosted by giving the newer Agile practitioners the knowledge to realise success in Agile software development through the critical success factors identified by other experienced Agile practitioners.

2.5 SYSTEMATIC LITERATURE REVIEW

Organisations and their environments change rapidly, as do critical success factors for software development (Siau et al., 2010). As a result, researchers must constantly work to identify the critical success factors that determine whether a project succeeds or fails. (Shehzad and Kausar, 2021). This study's systematic literature review was conducted to summarise the empirical findings on the critical success factors of Agile software development.

This structured approach was taken by similar studies investigating the critical success factors; thus, it is poised to be an appropriate approach. The section begins with background information required for the systematic literature review on the critical success factors theory and project success. Next, the background information on the SLR methodology is detailed, followed by a discussion on the findings of the SLR and concludes by detailing the research gap identified.

2.5.1 Background

2.5.1.1 Critical Success Factors Theory

Organisations and their environments change rapidly, as do critical success factors for systems development. Rockart and Crescenzi (1984) first created the Critical Success Factor approach to identifying and evaluating an organisation's performance, which (Bullen and Rockart, 1981) later improved and solidified. Bullen and Rockart (1981) define critical success factors as "the limited number of areas in which satisfactory outcomes will guarantee successful competitive performance for the organisation department or individual". Thus, for this study, the critical success factors are the aspects that must be present for an Agile project to succeed and ensure competitive advantage. This section will discuss the definition of success in Agile software development and the critical success factors identified in the existing literature.

2.5.1.2 Project Success In Agile Software Development

Software development project success is affected by several factors (Arcos-Medina and Mauricio, 2020), leading to the many definitions of what success in software development means (Mirnalini and Raya, 2010). Success can mean different things depending on the viewpoint and the stakeholder (Pereira et al., 2022). The other stakeholders in the Agile project may not always be satisfied with what satisfies one stakeholder. As Shenhar et al. (2001) said, "Success means different things to different people". That is why it is essential first to understand what success is in Agile software development projects before identifying the critical success factors.

Over time, the methods for determining project success and how to obtain it have changed (Tam et al., 2020). One of the more popular metrics today to evaluate project success is the Iron triangle (Pereira et al., 2022). The iron triangle says a project is successful if it is delivered on time and within scope and budget (Arcos-Medina and Mauricio, 2020). However, this definition may not always be applicable. Some projects have run behind schedule and over budget but have succeeded remarkably (Aldahmash, 2018).

Shenhar et al. (2001) argue that time and money alone cannot determine a project's success and that customer satisfaction should be considered. Consumer satisfaction relates to how the customer evaluates the end product's performance, including how well it adheres to a pre-defined set of goals (Tam et al., 2020). In Tam et al. (2020) study of the factors influencing the success of ongoing Agile software development projects, they also considered customer satisfaction as a measure of success along with time and cost. They stated that the recent definition of project success in software development has been placing emphasis on focusing on the customer; thus, they included customer satisfaction.
Adding this measure aligns the definition of project success with the principles defined in the Agile manifesto. Customer satisfaction is the highest priority in Agile development (Beck et al., 2001).

Chow and Cao (2008) researched the critical success factors in Agile software projects. They included quality as a measure of success in addition to time, cost and scope. Quality was defined as delivering a product that works properly, and scope was defined as meeting all customer requirements. Chiyangwa and Mnkandla (2017) emphasise that acceptance and understanding of Agile methodology, time, quality, budget, and scope are the central pillars for the success of an Agile project. Essentially, time, cost, scope, and quality significantly determine success, as does client satisfaction. It can be argued that the project will not be successful in the way that is intended if the consumer disapproves of the final product (Pereira et al., 2022).

Considering all the definitions and measures of success in Agile software development, the researcher has decided that for this research, the definition of success will be based on time, cost, quality, scope, and customer satisfaction. Thus, the elements of project success and the definitions by Chow and Cao (2008) and Shehzad and Kausar (2021) are summarised in Table 4.

Table 2.4: Elements of Project Success

Dimension	Element	Definition
Overall perceived level of success	Time	Delivering on time
	Cost	Delivering within budget
	Scope	Meeting all requirements and objectives
	Quality	Delivering a good working product
	Customer Satisfaction	Customer approval of delivered product

Source : (Chow and Cao, 2008, Shehzad and Kausar, 2021)

2.5.2 Systematic literature review methodology

A literature review is an essential component of academic research (Webster and Watson, 2002). It is essential because knowledge progress needs to be based on already completed work (Xiao and Watson, 2019). An efficient and well-conducted review provides a solid framework for knowledge expansion and the facilitation of theory building (Webster and Watson, 2002). There are several existing guidelines for literature review, and according to the specific objective, any of them may be helpful (Snyder, 2019). There are even more specific guidelines for conducting literature reviews in Information systems and software engineering (Webster and Watson, 2002), (Kitchenham, 2004), (Levy and Ellis, 2006), (Budgen and Brereton, 2006), (Bandara et al., 2011), (Wolfswinkel et al., 2013).

Kitchenham (2004) did a study to offer thorough guidelines for performing a systematic literature review suited for various software engineering studies. Software engineering is presented as a sub-discipline of software engineering by many authors, including (Chow and Cao, 2008), (Jiang and Eberlein, 2008), (Hoda et al., 2018) and (Dudhat and Abbasi, 2021).

Thus, this study conducted a systematic literature review using the guidelines by Kitchenham (2004). A systematic literature review is " *identifying, evaluating and interpreting all available research relevant to a particular research question, topic area, or phenomenon of interest"* (Kitchenham, 2004:1). In the author's report, they. The author suggests starting with a review protocol, which was done in this research

(Appendix A). After the protocol comes identifying resources, study selection, data extraction, data synthesis and report writing. The next sections will follow this order.

2.5.3 Search terms

("Agile software development" OR "Agile") AND ("critical success factors" OR "success factors" OR "success") AND ("projects" Or "South Africa" OR "software development" OR "software organisation")

2.5.4 Selection criteria

This section's inclusion and exclusion criteria were used to select research material for this systematic review.

2.5.1.3 Inclusion criteria

For a source to be included in the research, it had to meet the following criteria:

- Papers that identify the factors that influence Agile software development
- Papers that discuss Agile software development in a software development organisation
- Papers that discussed factors that contribute to the success of Agile software development
- Conference papers, journal articles, book chapters, thesis and dissertations are considered for review

2.5.1.4 Exclusion criteria

A source was excluded from the research if it met the following criteria:

- Papers with no explicit discussion about Agile
- Papers that do not provide information on factors contributing to Agile software development
- Papers not written in English
- Papers where the full text is not available to the researcher
- Duplicate papers (the same papers taken from different databases)
- Duplicate reports of the same study (the complete version will be selected)

2.5.5 Source Selection

The following data sources were selected to perform the search:

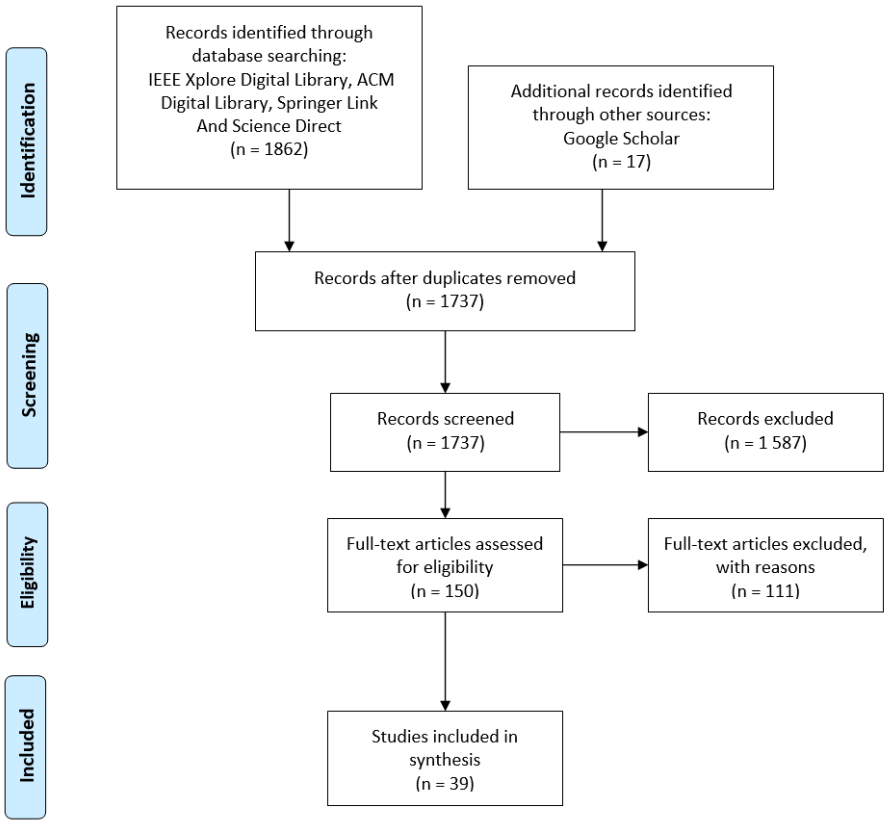
- IEEE Xplore Digital Library
- ACM Digital Library
- Springer Link
- Science Direct

These are all recognised research databases in the information technology field and were recommended by the Faculty of Engineering, Built Environment and Information Technology at the University of Pretoria. In addition, Google Scholar was used to find content missing in the databases and to do citation searches.

2.5.6 Prisma flowchart

As shown in Figure 4, a search using the search string was performed on the following databases: IEEE Xplore Digital Library, ACM Digital Library, Springer Link, and Science Direct. A total of 1862 articles were identified, along with 17 citation searches. Thus, there was a total of 1879 articles. Duplicate papers were removed, leaving 1737 articles. Screening by the title was conducted, leaving 384 full-text articles. An abstract screening was then performed on these papers, leaving 85. These articles were further screened using the selection criteria detailed in section 2.3. After the screening, 39 articles were left, and these were used to conduct the thematic analysis through the data extraction process, as shown in Appendix B.

Figure 2.4: Prisma Flowchart



2.5.7 Quality assessment

The quality of a Systematic Literature Review is equivalent to the quality of the publications evaluated (Zhou et al., 2015). Thus, performing a quality assessment of the studies to be included in this paper is essential. Kitchenham (2004) explains that a challenge in the quality assessment of papers is that there is no agreed-upon definition of study "quality". The authors suggest the method of the degree to which the research minimises bias and maximises external and internal validity. These aspects are defined as follows: Objectivity - whether the research is free of bias; reliability - the accuracy

and reliability of the research instruments used; Internal validity - whether the research was well structured, so data was collected from suitable sources; and External validity – determines if the findings can be predicted for subsequent occasions.

However, Oates (Oates) suggests that the assessment method based on objectivity, reliability, internal validity, and external validity is beneficial for positivist research, not interpretive research. (Keele, 2007) presents two checklists to assess the quality of qualitative and quantitative research based on reporting, rigour, credibility, and relevance. Thus, a researcher can use several methods to assess the quality of the primary studies to include in the research. To aid future literature reviews in selecting a method for quality assessment, Zhou et al. (2015) studied the practices of quality assessment in systematic literature reviews done in software engineering from 2011 to 2013. They identified and suggested the most often used standards and criteria, providing choices for researchers to consider when designing their quality evaluation.

Thus, the quality assessment in this systematic literature review will be done using a checklist adapted from the findings by assessment (Zhou et al., 2015), as shown in the table below.

Table 2.5: Quality Assessment Questions

Quality Assessment Question		Score		
		0	0.5	1
Reporting	Q1. Is there a clear statement of the research's aims (goals, objectives)?			
	Q2. Does the study answer the research question defined or present the results?			
Rigour	Q3. Are the metrics (methods, design, measures) used in the study clearly defined and justified?			
	Q4. Does the study provide a description and justification of the data analysis approaches?			
Credibility	Q5. Is the study replicable?			
	Q6. Do the researchers discuss any problems (limitations, threats) with the validity (reliability) of their results?			
Relevance	Q7. Is the value of the study evident?			
	Q8. Are conclusions, implications for practice, and future research reported suitably for its audience?			

Each paper was given a score for each question, either zero, 0.5 or one. Where zero means it did not meet the criteria or is not evident from the paper, 0.5 means it met part of the criteria partially with some possible issues, and one fully met the criteria.

For example, " Are conclusions, implications for practice, and future research reported suitably for its audience?". A paper would receive a one if it reported conclusions, implications for practice, and future research suitably for the audience. On the other hand, it would receive a zero if it did neither and a 0.5 if it only did it for conclusions, implications of practice or future practice but not for all.

This scoring procedure is recommended by Kitchenham (2004), and the paper needed to receive a minimum score of 0.5 on each of the questions to be included in the study.

2.5.8 Data extraction

After the papers' selection and quality assessment, relevant data were extracted to answer the research question. The following standard data were extracted from each paper:

- Title
- Publication Type
- Publication Year
- Publication Country
- Publication Continent
- Database source

Furthermore, the concepts in the paper that aid in answering the research question were extracted. These concepts, also called codes, were synthesised to categorise them into common themes. Appendix B shows how all this data was captured in the Excel spreadsheet.

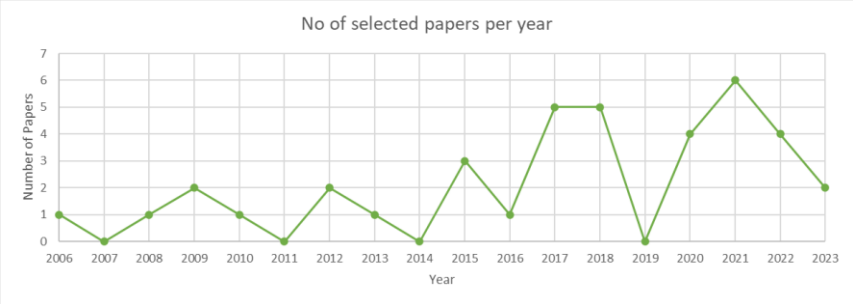
2.5.8.1 Results

The results of the included primary studies were summarised and collated quantitatively. The section presents the distribution of the studies by publication year and continent. The search results are also listed in a table indicating the main concepts found in each study.

2.5.8.2 Search results

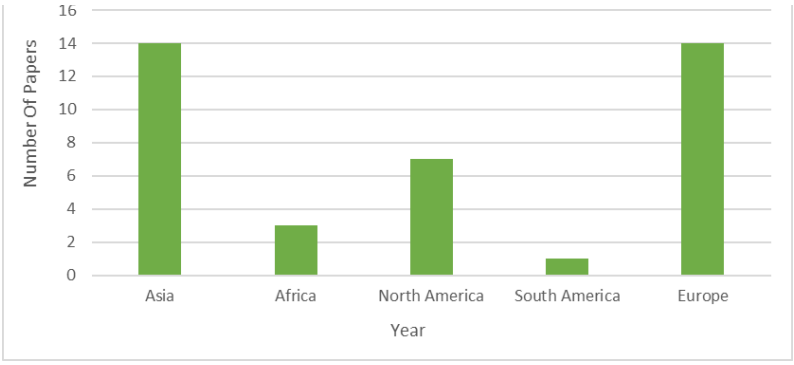
Considering the studies by year, it is observed that there has been an increasing number of publications from 2017 on the critical success factors in agile software development, except in 2019. The highest number of publications was in 2021, as shown in Figure 5.

Figure 2.5: Number of selected papers per year



The subsequent analysis was based on the publication by continent. Most publications are from Asia and Europe, with none in Antarctica and Australia. The three papers in Africa were all published in South Africa. Only 7% of the papers were written in South Africa and were quantitative studies. This further reinforces the need for further studies of agile critical success factors in South Africa and the use of a qualitative approach.

Figure 2.6: Number of selected papers per continent



2.5.8.3 Themes Identified in Literature.

The primary studies were analysed to identify the themes emerging from the studies. Earlier studies by Chow and Cao (2008), Misra et al. (2006) and Misra et al. (2009) grouped the themes into categories named Organisational, People, Process, Technical and Project. However, more recent papers (Arcos-Medina and Mauricio, 2020, Hummel and Epp, 2015, Shakya and Shakya, 2020) divide the people factor into team and customer categories. Thus, it was poised to use the categories catering to earlier and most recent literature, leading to the themes in Table 6.

Most of the studies of the papers covered more than one dimension, while a few focused their research on a specific dimension, with organisational being the most addressed. The list of all the primary studies and the categories that were covered is shown in Appendix C.

Table 2.6: Themes identified in the literature.

Themes	Definition	Source
1. Organisational	This category covers factors relating to the organisational structure, organisational environment and administrative climate of the company.	(Arcos-Medina and Mauricio, 2020, Chow and Cao, 2008, Hummel and Epp, 2015, Misra et al., 2006, Shakya and Shakya, 2020)
2. Team	This category covers factors relating to the people who manage and execute the project.	(Arcos-Medina and Mauricio, 2020, Hummel and Epp, 2015, Shakya and Shakya, 2020)
3. Customer	This category covers factors relating to the people who sponsor the project or will use the product.	(Arcos-Medina and Mauricio, 2020, Hummel and Epp, 2015, Shakya and Shakya, 2020)
4. Process	This category covers factors relating to how project activities are carried out.	(Arcos-Medina and Mauricio, 2020, Hummel and Epp, 2015, Shakya and Shakya, 2020)
5. Technical	This category includes factors relating to the tools, technologies, or techniques used in the project.	(Arcos-Medina and Mauricio, 2020, Chow and Cao, 2008, Hummel and Epp, 2015, Misra et al., 2006, Shakya and Shakya, 2020)
6. Project	This category covers relating to the project parameters	(Arcos-Medina and Mauricio, 2020, Chow and Cao, 2008, Hummel and Epp, 2015, Misra et al., 2006, Shakya and Shakya, 2020)

2.5.9 Quality Evaluation of Primary Studies

Eight questions on relevance, credibility, rigour and reporting were used to assess the quality of the primary studies, as mentioned in the quality assessment section. Table 7 displays the quality assessment scores. All the papers included in the final data synthesis scored at least 0.5 in each question.

Table 2.7: Quality assurance scores

No	Citation	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
1	(Aldahmash et al., 2017)	1	1	1	1	0.5	0.5	1	1
2	(Aldahmash et al., 2018)	1	1	1	1	1	1	1	1
3	(Aldahmash, 2018)	1	1	1	1	1	1	1	1
4	(Alvarez & Sánchez, 2022)	1	1	1	1	1	1	1	1
5	(Arcos-Medina & Mauricio, 2020),	1	1	1	1	1	1	1	1
6	(Asnawi, 2012)	1	1	1	1	1	1	1	1
7	(Azhar & Abdullah, 2022)	1	1	1	1	1	1	1	1
8	(Chiyangwa, 2017)	1	1	1	1	1	1	1	1
9	(Chow & Cao, 2008)	1	1	1	1	1	1	1	1
10	(Cucolaş & Russo, 2023)	1	1	1	1	1	1	1	1
11	(Dieteren, 2022)	1	1	1	1	1	1	1	1
12	(Ghayyur et al., 2018)	1	1	1	1	1	1	1	1
13	(Hamdani & Butt, 2017)	1	1	1	1	1	1	1	1
14	(Hummel & Epp, 2015)	1	1	1	1	1	1	1	1
15	(Jintian et al., 2022)	1	1	1	1	1	1	1	1
16	(Kelle et al., 2015)	1	1	1	1	1	1	1	1
17	(Kulathunga & Ratiyala, 2018)	1	1	1	1	1	1	1	1
18	(Misra et al., 2006)	1	1	1	1	1	1	1	1
19	(Misra et al., 2009)	1	1	1	1	1	1	1	1
20	(Nguyen, 2016)	1	1	1	1	1	1	1	1
21	(Noteboom et al., 2021),	1	1	1	1	1	1	1	1
22	(Peters, 2020)	1	1	1	1	1	1	1	1
23	(Qatanani et al., 2021)	1	1	1	1	1	1	1	1
24	(Riaz et al., 2018)	1	1	1	1	1	0.5	1	1
25	(Russo, 2021)	1	1	1	1	1	1	1	1
26	(Shakya & Shakya, 2020)	1	1	0.5	0.5	1	1	1	1

No	Citation	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
27	(Shameem et al., 2020)	1	1	1	1	1	1	1	1
28	(Shameem et al., 2023)	1	1	1	1	1	1	1	1
29	(Shehzad & Kausar, 2021)	1	1	1	1	1	1	1	1
30	(Stankovic et al., 2013)	1	1	1	1	1	1	1	1
31	(Stelzmann et al., 2010)	1	1	1	1	1	1	1	1
32	(Tam, Moura, et al., 2020)	1	1	1	1	1	1	1	1
33	(Aldahmash et al., 2017)	1	1	1	1	1	1	1	1
34	(Anjani et al., 2021)	1	1	1	1	1	1	1	1
35	(Kouzari et al., 2015)	1	1	1	1	1	1	1	1
36	(Campanelli et al., 2017)	1	1	1	1	1	1	1	1
37	(Shameem et al., 2017)	1	1	1	1	1	1	1	1
38	(Tsirakidis et al., 2009)	1	1	1	1	1	1	1	1
39	(Perera and Perera, 2019)	1	1	1	1	1	1	1	1

2.5.10 DISCUSSION

This section of the systematic literature review discusses the critical success factors identified in the existing literature. The discussion is a brief, high-level discussion as the main aim of this research is not the SLR; however, the SLR was used to gather and understand what is in the existing literature. Doing this allowed the researcher to ascertain if the findings from the case study supported what existed at the point the SLR was conducted and if any new critical success factors were identified.

The factors identified in the SLR are listed in Table 8 and grouped by category. The discussion is organised by the categories identified in the SLR and mentions the factors in each category and their related attributes.

Table 2.8: Critical success factors identified in the SLR

Factor	Attributes	Freq	Source
Organisational Factors			
Management support	• Strong support from executives, team leaders, and project managers • Management buy-in into the project. • Acceptance of the project • A committed sponsor or manager. • Consistent support • Clear vision from leadership • Product owner involvement	24	(Chow and Cao, 2008) ,(Aldahmash et al., 2018) (Aldahmash et al., 2017) (Asnawi, 2012), (Ghayyur et al., 2018), (Arcos-Medina and Mauricio, 2020),(Peters, 2020), (Kulathunga and Ratiyala, 2018), (Shehzad and Kausar, 2021), (Noteboom et al., 2021), (Shakya and Shakya, 2020), (Stankovic et al., 2013), (Hummel and Epp, 2015), (Qatanani et al., 2021),(Hamdani and Butt, 2017),(Wagener, 2012), (Russo, 2021), (Dieteren, 2022), (Azhar and Abdullah, 2022), (Alvarez and Sánchez, 2022)
Organisational culture	• Promoting corporation, teamwork and collaboration over rank or hierarchy • Openness and Transparency • A culture of trust, loyalty, and commitment • result-oriented • risk-taking organisation • A dynamic and fast-altering firm • Democratic culture • Allowed to take the initiative.	21	(Chow and Cao, 2008) (Shameem et al., 2020) (Shehzad and Kausar, 2021) (Aldahmash et al., 2018) (Stankovic et al., 2013) (Asnawi, 2012) (Peters, 2020), (Ghayyur et al., 2018), (Kulathunga and Ratiyala, 2018), (Riaz et al., 2018), (Shehzad and Kausar, 2021), (Misra et al., 2006),(Chiyangwa, 2017), (Hummel and Epp, 2015),(Qatanani et al., 2021),(Misra et al., 2009),(Wagener, 2012), (Shakya and Shakya, 2020), (Shameem et al., 2023),(Dieteren, 2022), (Azhar and Abdullah, 2022)
Societal culture	• Cultural aspects • a society with low power distance (a non-hierarchical organisation in which every decision is not made from the Top) • a society with low uncertainty avoidance (not sceptical about new methods) • inherent regional culture	5	(Asnawi, 2012), (Tam et al., 2020), (Stankovic et al., 2013), (Misra et al., 2006),(Misra et al., 2009)
Agile-friendly work environment	• Communication and collaboration layout • pair programming accommodation • Work-location • A high degree of agility	13	(Peters, 2020), (Nguyen, 2016), (Kulathunga and Ratiyala, 2018), (Ghayyur et al., 2018), (Riaz et al., 2018), (Shehzad and Kausar, 2021), (Chiyangwa, 2017), (Qatanani et al., 2021) (Hummel and Epp, 2015),(Misra et al., 2009), (Wagener, 2012), (Cucolaş and Russo, 2023), (Azhar and Abdullah, 2022)
Decision time	• Short decision time • Organisational support for the decision-making of developers	4	(Misra et al., 2006),(Misra et al., 2009), (Peters, 2020), (Shehzad and Kausar, 2021)
Collaboration	• use of social technologies and tools	8	(Stelzmann et al., 2010), (Kelle et al., 2015) (Chow and Cao, 2008) (Ghayyur et al., 2018) (Aldahmash et al., 2018) (Aldahmash et al., 2017),(Shehzad and Kausar, 2021),(Shakya and Shakya, 2020)
Effective communication	• Direct communication between customers and the development team • Informal communication between team members • face-to-face communication	12	(Peters, 2020),(Nguyen, 2016), (Ghayyur et al., 2018), (Shehzad and Kausar, 2021), (Noteboom et al., 2021),(Chiyangwa, 2017),(Wagener, 2012), (Shameem et al., 2023),(Dieteren, 2022), (Azhar and Abdullah, 2022),(Alvarez and Sánchez, 2022), (Stelzmann et al., 2010)
Training	• Enhancement of team technical skills • Education and Continuous Learning • Knowledge sharing • Mentoring • Formal and informal training • Empowerment of the project team • Guidance through discussions	15	(Asnawi, 2012), (Peters, 2020), (Aldahmash et al., 2018), (Aldahmash, 2018), (Nguyen, 2016),(Ghayyur et al., 2018), (Shehzad and Kausar, 2021), (Noteboom et al., 2021),(Shakya and Shakya, 2020), (Tam et al., 2020), (Misra et al., 2006), (Qatanani et al., 2021), (Dieteren, 2022), (Azhar and Abdullah, 2022),(Alvarez and Sánchez, 2022)
Reward system appropriate for agile methodologies	• recognition of good work • incentive programs	7	(Arcos-Medina and Mauricio, 2020), (Peters, 2020), (Nguyen, 2016), (Ghayyur et al., 2018), (Shehzad and Kausar, 2021), (Qatanani et al., 2021),(Wagener, 2012)

Factor	Attributes	Freq	Source
Team Factors			
Team motivation and commitment	• Personal interest in the project • Right attitude (team and individual) • willingness to try new methods. • Team autonomy • Availability of team members • Team motivation • Willingness to work. • dedicated	11	(Chow and Cao, 2008) (Shameem et al., 2020),(Asnawi, 2012), (Peters, 2020), (Ghayyur et al., 2018),(Wagener, 2012),(Jintian et al., 2022), (Nguyen, 2016), (Noteboom et al., 2021), (Hummel and Epp, 2015),(Dieteren, 2022)
Team capabilities	• Strong technical skills • Knowledge about the system • Knowledge about the methods • High capabilities • Clear domain knowledge • real-world experience in the technology domain	18	(Ghayyur et al., 2018) (Chow and Cao, 2008) (Aldahmash et al., 2018) (Aldahmash et al., 2017) (Stankovic et al., 2013) (Tam et al., 2020), (Peters, 2020), (Nguyen, 2016), (Misra et al., 2006), (Shehzad and Kausar, 2021), (Noteboom et al., 2021), (Shakya and Shakya, 2020), (Stankovic et al., 2013), (Aldahmash et al., 2017), (Qatanani et al., 2021),(Wagener, 2012),(Russo, 2021), (Azhar and Abdullah, 2022)
Individual Characteristics	• having collaborative attitude • honesty, trust • sense of responsibility, and eagerness to learn. • Interpersonal skills • Communication skills	11	(Nguyen, 2016), (Kulathunga and Ratiyala, 2018), (Tam et al., 2020), (Misra et al., 2006), (Aldahmash et al., 2017), (Hummel and Epp, 2015), (Misra et al., 2009),(Peters, 2020), (Russo, 2021),(Dieteren, 2022),(Alvarez and Sánchez, 2022)
No multiple teams	• Teams are not separate	3	(Peters, 2020), (Chiyangwa, 2017), (Chow and Cao, 2008)
Team size	• Small team	12	(Chow and Cao, 2008) (Shameem et al., 2020) (Ghayyur et al., 2018), (Peters, 2020), (Shehzad and Kausar, 2021), (Noteboom et al., 2021),(Shakya and Shakya, 2020), (Chiyangwa, 2017), (Qatanani et al., 2021),(Hamdani and Butt, 2017),(Wagener, 2012),(Dieteren, 2022)
Agile Mindset	• understanding of agile methodology • personnel aligned as per the Agile manifesto	5	(Asnawi, 2012) (Ghayyur et al., 2018), (Shehzad and Kausar, 2021),(Dieteren, 2022),(Alvarez and Sánchez, 2022)
Coherent, Self-organizing team	• a sense of ownership • Team thinking, • Team taking responsibility	11	(Chow and Cao, 2008) (Shameem et al., 2020) (Peters, 2020),(Qatanani et al., 2021), (Chiyangwa, 2017),(Russo, 2021), (Jintian et al., 2022),(Stelzmann et al., 2010),(Peters, 2020), (Stankovic et al., 2013), (Misra et al., 2006),
Manager with Agile suitable management style	• Light touch management style • Adaptive management • skilled in agile processes • transformational leadership • scrum master leadership	8	(Peters, 2020), (Riaz et al., 2018), (Shehzad and Kausar, 2021), (Noteboom et al., 2021), (Qatanani et al., 2021), (Wagener, 2012),(Russo, 2021),(Jintian et al., 2022)

Factor	Attributes	Freq	Source
Customer Factors			
Customer Involvement	• Customer commitment (available and highly active) • Customers work closely with the development team. • Customer support • Customers have full authority. • viewing themselves as responsible components • Customer collaboration	21	(Chow and Cao, 2008) (Shameem et al., 2020) (Ghayyur et al., 2018) (Aldahmash et al., 2018) (Aldahmash et al., 2017) (Stankovic et al., 2013) (Asnawi, 2012) (Peters, 2020) (Tam et al., 2020), (Nguyen, 2016), (Kulathunga and Ratiyala, 2018),(Shakya and Shakya, 2020), (Stankovic et al., 2013), (Aldahmash et al., 2017), (Hummel and Epp, 2015), (Qatanani et al., 2021),(Misra et al., 2009),(Wagener, 2012), (Shameem et al., 2023), (Azhar and Abdullah, 2022)
Good customer relationship	• -	5	(Peters, 2020), (Noteboom et al., 2021),(Shakya and Shakya, 2020), (Chiyangwa, 2017),(Qatanani et al., 2021)
Customer satisfaction	• conformance to requirements • fitness for use	3	(Kulathunga and Ratiyala, 2018),(Shakya and Shakya, 2020), (Misra et al., 2009)
Process Factors			
Agile-oriented project management process	• Rapid configuration • Progress tracking • Agile-oriented configuration management	11	(Aldahmash et al., 2017) (Stankovic et al., 2013) (Peters, 2020), (Nguyen, 2016), (Kulathunga and Ratiyala, 2018), (Ghayyur et al., 2018), (Shakya and Shakya, 2020), (Stankovic et al., 2013), (Chiyangwa, 2017),(Dieteren, 2022), (Azhar and Abdullah, 2022)
Agile-oriented requirement management process	• Customer involvement in requirements elicitation • Prioritisation of requirements	7	(Peters, 2020), (Nguyen, 2016), (Noteboom et al., 2021),(Shakya and Shakya, 2020),(Chiyangwa, 2017), (Shameem et al., 2023),(Alvarez and Sánchez, 2022)
Rapid feedback	• Short iterations • Face-to-face feedback session	5	(Ghayyur et al., 2018),(Hummel and Epp, 2015),(Qatanani et al., 2021), (Hamdani and Butt, 2017), (Shameem et al., 2023)
Proper project planning	• Documented plans • quantitative performance measures	6	(Nguyen, 2016),(Shakya and Shakya, 2020),(Misra et al., 2006), (Aldahmash et al., 2017),(Hummel and Epp, 2015),(Wagener, 2012)
Proper project definition	• Accurate effort estimation • Realistic expectations • Clear goals • Signed-of scope. • Requirements are fed into a product backlog before sprint inception. • Acceptance criteria • Product delivery parameters	8	(Kulathunga and Ratiyala, 2018), (Ghayyur et al., 2018), (Shehzad and Kausar, 2021), (Noteboom et al., 2021), (Stankovic et al., 2013),(Chiyangwa, 2017), (Aldahmash et al., 2017), (Azhar and Abdullah, 2022)
Risk analysis	• Analysing risk prior to project commencement	3	(Peters, 2020), (Kulathunga and Ratiyala, 2018), (Ghayyur et al., 2018), (Chow and Cao, 2008)
Agile software development standards	• Simple design influences • Simple code • Properly defined coding standards upfront • inexpensive refactoring • Only necessary documentation • Sharing the code base amongst programmers	14	(Aldahmash et al., 2018) (Aldahmash et al., 2017) (Stankovic et al., 2013),(Peters, 2020), (Kulathunga and Ratiyala, 2018), (Ghayyur et al., 2018), (Shehzad and Kausar, 2021),(Shakya and Shakya, 2020), (Stankovic et al., 2013),(Misra et al., 2006), (Chiyangwa, 2017),(Hamdani and Butt, 2017),(Wagener, 2012),(Stelzmann et al., 2010)
Continues Delivery strategy	• Rapid delivery • incremental delivery at the end of each sprint. • Each iteration delivers a working product or prototype. • Delivering the most important features first	9	(Chow and Cao, 2008) (Aldahmash et al., 2018) (Aldahmash et al., 2017) (Stankovic et al., 2013) (Peters, 2020), (Kulathunga and Ratiyala, 2018), (Stankovic et al., 2013),(Stelzmann et al., 2010)
Honouring a regular working schedule	• No overtime	5	(Chow and Cao, 2008), (Arcos-Medina and Mauricio, 2020), (Nguyen, 2016), (Qatanani et al., 2021), (Wagener, 2012)

Factor	Attributes	Freq	Source
Technical Factors			
Advanced tools and technologies	• Rich technological infostructure • Suitable development tools • Suitable Communication Media • Progress Tracking tools	6	(Nguyen, 2016), (Aldahmash, 2018), (Shehzad and Kausar, 2021),(Hamdani and Butt, 2017), (Shameem et al., 2023), (Azhar and Abdullah, 2022)
Testing	• Automated testing • Continuous integration testing • Regression Testing • User acceptance testing • Executable test cases • Linking test cases and requirements	6	(Peters, 2020),(Nguyen, 2016), (Misra et al., 2006),(Wagener, 2012), (Azhar and Abdullah, 2022),(Stelzmann et al., 2010)
Project Factors			
Project Type	• Emergent requirements • Variable scope project	5	(Peters, 2020), (Kulathunga and Ratiyala, 2018), (Noteboom et al., 2021), (Misra et al., 2006), (Azhar and Abdullah, 2022)
Project Nature	• Non-life critical	4	(Stankovic et al., 2013) (Chow and Cao, 2008), (Peters, 2020), (Chiyangwa, 2017)
Adaptive project schedule	• Dynamic • Accelerated	5	(Stankovic et al., 2013) (Chow and Cao, 2008), (Peters, 2020), (Nguyen, 2016), (Kulathunga and Ratiyala, 2018)

2.5.8.4 Organisational Factors

The Organisation significantly impacts a project's success since it shapes the environment in which the team developing the software operates, the culture and the overall mindset (Azhar and Abdullah, 2022). This category includes the following factors: management support, organisational culture, societal culture, agile-friendly work environment, decision time, collaboration, effective communication, training, and a reward system.

The management support factor has the following attributes: strong support from executives, team leaders, product owners, and project managers, management buy-in, acceptance of the project, committed sponsor or manager, consistent support, and clear vision from leadership. The organisational culture comprises promoting corporation, teamwork, and collaboration over rank or hierarchy, openness and transparency, a culture of trust, loyalty, and commitment, a result-oriented, risk-taking organisation and a dynamic and fast-altering firm (Chiyangwa and Mnkandla, 2017, Misra et al., 2006, Peters, 2020, Riaz et al., 2018).

As with any other human activity, inherent local culture may significantly impact software development, making it essential for success in agile software

development projects (Misra et al., 2009). The societal culture factor has the following attributes: cultural aspects, a society with low power distance, a society with low uncertainty avoidance, and inherent regional culture. An Agile-friendly work environment is key to agile success, and this is one with a communication and collaboration layout, pair programming accommodation, work location, and a high degree of agility (Cucolaş and Russo, 2023, Ghayyur et al., 2018, Kulathunga and Ratiyala, 2018, Peters, 2020).

In Agile software development, successful teams are usually left to their own, enabling them to make their own decisions (Misra et al., 2009). The decision-time factor's attributes are short decision times and organisational support for the decision-making of the team members. The collaboration factor entails using social technologies and tools to communicate. In agile development, the communication factor plays a critical role in the success of a project, which involves direct communication between customers and the development team, informal communication between team members, and face-to-face communication (Aldahmash et al., 2017).

In agile software development, people should be enthusiastic about exchanging information with one another and constantly learn (Misra et al., 2006). Knowledge sharing increases the chances of agile practices being successful. Training attributes include enhancement of team technical skills, education and continuous learning, knowledge sharing, mentoring, discussion guidance, formal and informal training, and empowerment of the project team (Alvarez and Sánchez, 2022, Chiyangwa, 2017). The reward system factor has the following attributes: recognition of good work and incentive programs.

2.5.8.5 Team Factors

Team factors describe the required properties and attributes of the team members to get the job done (Hummel and Epp, 2015). The factors in this category include team motivation and commitment, team capability, individual characteristics, no multiple teams, team size, agile mindset, coherent self-organizing team, and having a manager with an Agile-suitable management style.

Motivated team members should be the foundation of agile software development (Aldahmash et al., 2017). The team motivation and commitment factor can be described as having a personal interest in the project, having the right attitude (team and individual), team autonomy, availability of team members, team motivation, and willingness to work and try new methods (Dieteren, 2022, Jintian et al., 2022, Noteboom et al., 2021). Working agile includes a great deal of informal communication and implicit knowledge; the more robust the team spirit, the more excellent the team's performance. (Stelzmann et al., 2010). The team's capabilities include strong technical skills, knowledge about the system and methods, high capabilities, explicit domain knowledge, and real-world experience in the technology domain (Chow and Cao, 2008, Stankovic et al., 2013, Tam et al., 2020, Wagener, 2012).

The individual characteristics factor include a collaborative attitude, honesty, trust, a sense of responsibility, eagerness to learn, and interpersonal and communication skills (Radhakrishnan et al., 2021). The team size factor refers to the project having a small team. Teams are often small to facilitate collaboration and communication (Wagener, 2012). The agile mindset factor is attributed to understanding the agile methodology and aligned personnel per the Agile manifesto (Dieteren, 2022).

A coherent, self-organizing team has a sense of ownership and team thinking and takes responsibility (Jintian et al., 2022, Russo, 2021). Lastly, having a manager with an Agile suitable management style means one with a light touch management style, adaptive management, skilled in agile processes, and transformational leadership (Noteboom et al., 2021, Russo, 2021).

2.5.8.6 Customer Factors

The customer factor category encompasses those factors relating to the people who sponsor the project or will use the product (Arcos-Medina and Mauricio, 2020). The customer's role calls for the customer or a customer representative to be involved frequently throughout the project (Azhar and Abdullah, 2022). The factors identified

in this category are customer involvement, good customer relationship and customer satisfaction.

Aldahmash et al. (2017) state that customer involvement is crucial during agile software development. It includes several attributes such as customer commitment is seen by them being available and highly active, customers working closely with the development team and customer support, customers having full authority and viewing themselves as responsible components and customer collaboration (Aldahmash, 2018, Asnawi, 2012, Chow and Cao, 2008, Shameem et al., 2020, Tam et al., 2020).

A good customer relationship is healthy between the organisation, project team, and customer (Qatanani et al., 2021). Customer satisfaction means the customer requirements have been met, and the product produced is fit for use (Misra et al., 2009, Shakya and Shakya, 2020).

2.5.8.7 Process Factors

The whole process lifecycle should be agile for success in agile software development, from requirements elicitation to management (Hamdani and Butt, 2017). This category includes the following factors: agile-oriented project management process, Agile-oriented requirement management process, rapid feedback, proper project planning, proper project definition, risk analysis, agile software development standards, continuous delivery strategy and honouring a regular working schedule.

The agile-oriented requirement management process means rapid configuration, progress tracking, and agile-oriented configuration management. (Aldahmash et al., 2017, Dieteren, 2022) The agile-oriented requirement management process means customers are involved in eliciting and prioritizing requirements (Alvarez and Sánchez, 2022).

The rapid feedback factor refers to having short iterations to deliver the output and face-to-face feedback sessions (Qatanani et al., 2021). Aldahmash et al. (2017) state that it is essential to launch a project with a project plan that is correctly sized. Proper project planning entails documenting plans and having quantitative performance measures to assess progress and success (Misra et al., 2006, Shakya and Shakya, 2020). This factor aligns with the proper project definition factor, which requires an accurate effort estimation and setting realistic estimations (Kulathunga and Ratiyala, 2018). The risk analysis factor refers to analysing risk before project commencement.

Continuous delivery is critical to agile software development and means rapid delivery, incremental delivery at the end of each sprint; each iteration delivers a working product or prototype and the most important features first (Aldahmash et al., 2018, Chow and Cao, 2008, Peters, 2020, Stankovic et al., 2013). The final factor in this category is honouring a regular working schedule without overtime.

2.5.8.8 Technical Factors

This category has the least factors, with only three identified critical success factors. These factors include using advanced tools and technologies, testing and agile software development techniques.

If the technology and tools used in an Agile software project are outdated or unsuitable, it will lead to project failure, just as misusing Agile methodologies will (Aldahmash et al., 2017). Using advanced tools and technologies has the following attributes: having a rich technological info structure, suitable development tools, suitable communication media and progress tracking tools (Aldahmash et al., 2017, Hamdani and Butt, 2017, Shameem et al., 2023).

The testing factor includes different testing methods, such as continuous integration testing, regression testing and user acceptance testing. In addition, the testing should be automated with executable test cases linked to requirements (Azhar and Abdullah, 2022, Wagener, 2012). The agile software development techniques factor has several attributes, including simple design influences, simple code, properly

defined coding standards upfront, inexpensive refactoring, having only necessary documentation, and sharing the code base amongst programmers (Chiyangwa and Mnkandla, 2017, Ghayyur et al., 2018)

2.5.8.9 Project Factors

Project factors include the project type, project nature and adaptive project schedule.

The project type factor refers to a project with emergent requirements and variable scope (Noteboom et al., 2021). Past research has shown that in agile projects, the project scope can be changed during the project implementation by adding new requirements (Kulathunga and Ratiyala, 2018). The nature of the project needs to be non-life-critical software, yet it may be corporate mission-critical software (Stankovic et al., 2013). Lastly, an adaptive project schedule with dynamic and accelerated project schedules is more likely to succeed (Peters, 2020).

Thirty-four critical success factors were identified in the systematic literature review. The factors were grouped into these categories: organisational, team, customer, process, technical and project. The factors will be used to develop the framework for the next section.

2.5.11 Summary and Gap in Literature

Thirty-five critical success factors were identified in the literature review. However, not all factors were listed by all references, with some having as few as three references. A study conducted in Nepal by Jintian et al. (2022) noted that support from top-level management was not a critical success factor. However, in this literature review, 24 other studies in different countries presented it as a CSF. Thus, is it only applicable to Nepal, and are these factors that Agile software practitioners in South Africa need not be concerned with? Peters (2020) study conducted in South Africa identified two additional CSFs of Agile software development: test automation

and cloud computing. These factors were not identified in the other studies and therefore beg the question: is it just applicable in the South African context? Future research must gather data from a different county to get a different perspective (Tam et al., 2020)

Additionally, the rankings are inconsistent. According to Aldahmash (2018), the highest-ranked factor is communication, followed by organisational culture. Nguyen (2016) states that the most prevalent themes are strong customer involvement, good agile project management processes, engineering techniques and practices and good technologies. However, they suggested that future work may research the CSF of Agile software development in other countries, suggesting that the ranking may vary in different countries. Peters (2020) mentions that in South Africa, there is a gap between the importance and performance of the critical success factors, which can be minimised by organisations emphasising the importance of the critical success factors that are integral to agile project success. This gap highlights the need to prioritise the critical success factors in South Africa.

2.6 THEORETICAL FRAMEWORK

A theoretical framework in research is a framework that can support or hold a theory for the research project and acts as the foundation for carrying out specific research (Chiyangwa and Mnkandla, 2017). The goal of using theory in the early phases of interpretive case studies is to establish an initial theoretical framework that incorporates prior knowledge and provides a sound theoretical foundation for the topics and methodology of the preliminary empirical research (Walsham, 1995).

The main theories utilized in research to describe the phenomenon of IS success within organisations include the Theory of Reasoned Action (TRA), the Technology Acceptance Model (TAM), the Theory of Planned Behaviour (TPB), the technology-organisation-environment (TOE) and Unified Theory of Acceptance and Use of Technology (UTAUT). However, no comprehensive framework exists to discover and create insights into all significant CSFs for agile development and their related constructs, which are extremely important to the software development community and organisations (Chiyangwa and Mnkandla, 2017).

In the study conducted by Chiyangwa and Mnkandla (2017), they aimed to find the most suitable theoretical framework that may be modified to model the essential elements of agile software development projects' success. After a thorough analysis, they concluded that it is the Unified Theory of Acceptance and Use of Technology (UTAUT). Therefore, it was modified to fit critical success factors for agile methodologies. The author grouped the critical success factors into technical, organisational, social, people, process, cultural, and political factors.

Several researchers on the critical success factors in agile software development do not employ a specific framework but rather synthesise existing frameworks. Constructs from TAM, TRA, TPB, and UTAUT were used by Chow and Cao (2008) and Misra et al. (2009) to identify the CSFs for agile software development projects. In these studies and several others (Aldahmash et al., 2017, Aldahmash et al., 2018, Aldahmash, 2018, Chiyangwa and Mnkandla, 2017, Chow and Cao, 2008, Nasehi, 2013, Stankovic et al.,

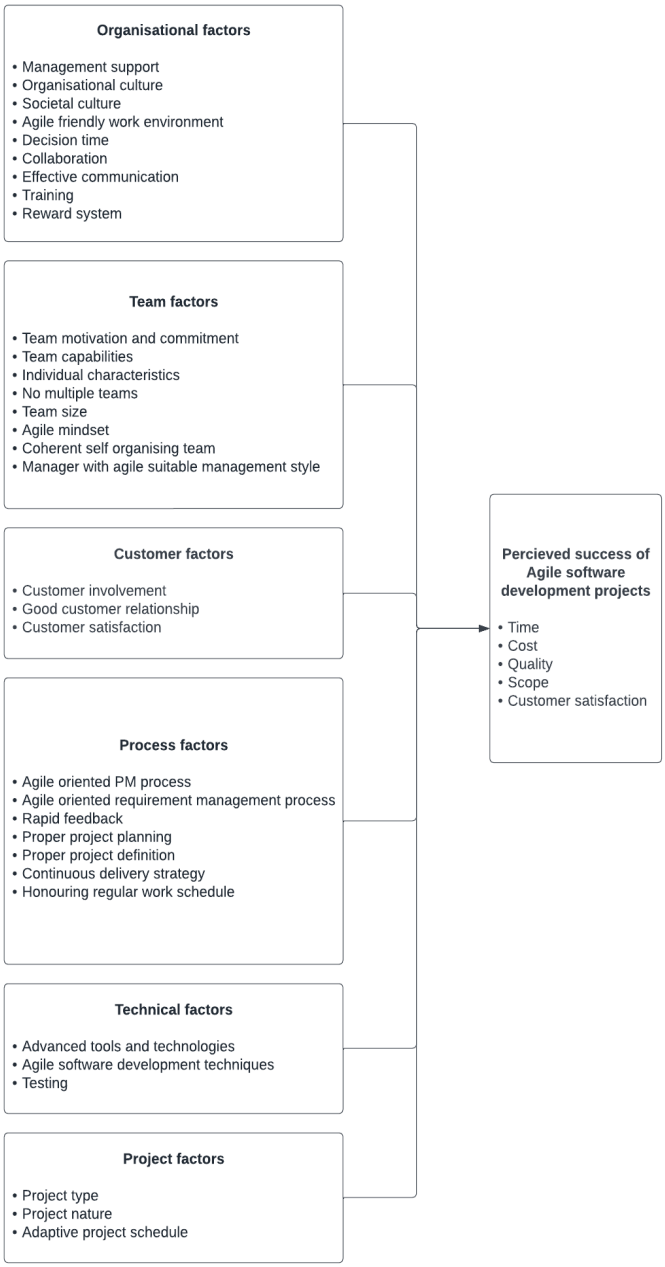
2013), they categorised the critical success factors into five categories: organisational, people, process, technical and project.

However, other studies, such as those by Shakya and Shakya (2020) and Hummel and Epp (2015), split the people factors into team and customer. Shakya and Shakya (2020) state that they revised an existing conceptual framework to make it reliable to Nepal. This implies that the categories for the CSF might not always have the same categories and groupings depending on the practitioners' perceptions within a geographical region, such as a country. Tona et al. (2019) aimed to identify the success factors specific to scrum. They stated that based on their studies, they know that team, human, project, and organisational factors directly impact the execution and implementation of Scrum and, as a result, the project's success or failure.

In interpretive research, it is preferable to maintain a high level of openness to the field data and a willingness to revise early theories (Walsham, 1995). Thus, it is poised that splitting the people factors into customer and team factors would be beneficial not to generalise “people” as the customer and team factors differ since the aim is to expand on the initial theory or generate a new theory based on the analysis.

Therefore, considering the approach taken by existing literature, a revised theoretical framework was constructed, as shown in Figure 7. This framework was based on the research model by Chow and Cao (2008) the later revisions by Tona et al. (2019) and Shakya and Shakya (2020) in their models and the Unified Theory of Acceptance and Use of Technology (UTAUT).

Figure 2.7: Research Theoretical Framework



2.7 CHAPTER SUMMARY

This chapter discussed the current research on areas related to the research topic. A literature review was done, and the findings were discussed in the following sections: software development, Agile methodologies, Agile software development and critical success factors. The software development lifecycle and traditional methodologies were discussed under software development, with waterfall as an example. Under Agile methodologies, the Agile Manifesto was discussed along with the most popular methodology- Scrum. The Agile software development section contained a comparison of Agile and traditional software development and the impact of Agile on software development. The critical success factors section discussed the definition of success in Agile software development and identified critical success factors in the literature. Lastly, a systematic literature was identified where 33 factors were identified from existing literature.

3 METHODOLOGY

This chapter details the underlying research philosophy in this study, outlines the research methodology selected, and details the data collection methods chosen.

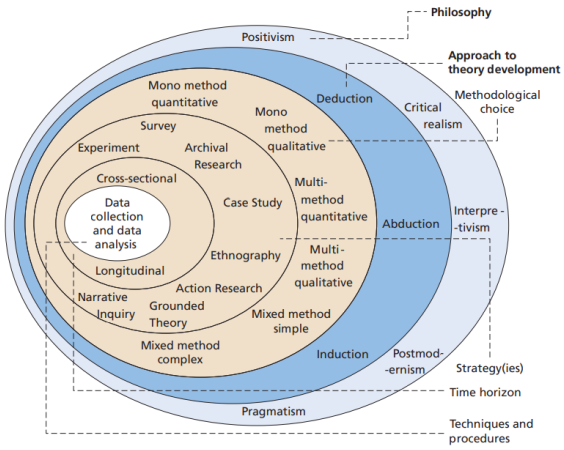
3.1 INTRODUCTION

The methodology in a study combines research strategies and the core data generation mechanisms a research project uses (Oates, 2006). It reflects the logic of the theory-generating process, the procedural framework within which the research is carried out (Mohajan, 2018). The research methodology is a way to systematically solve the research problem (Goddard and Melville, 2004). Therefore, the research was carefully planned and designed to answer the research question by collecting appropriate data. The sections of the chapter include research design, data collection, sampling, data analysis, ethics, and a conclusion summarizing the chapter.

3.2 RESEARCH DESIGN

The research design can be defined as "the conceptual structure within which research is conducted; it constitutes the blueprint for the collection, measurement and analysis of data" (Goddard and Melville, 2004:48). The research design was guided by the 'Research Onion' by Saunders et al. (2007), shown in Figure 8 and considers the need for qualitative research in this field as identified in the discussion of the problem statement.

Figure 3.1: The 'Research Onion'



Source: (Saunders et al., 2007:134)

3.3 THE RESEARCH PHILOSOPHY

The research philosophy is a" *system of beliefs and assumptions about the development of knowledge*" (Saunders et al., 2007:5). The first layer of the 'research onion' shows the different classifications based on philosophical assumptions. These are critical realism, positivism, interpretivism, pragmatism and postmodernism. Perren and Ram (2004) state that starting with outlining the philosophical paradigm of research leads researchers to consider the broader philosophical and epistemological consequences of their choices. The author explains that a researcher must decide the paradigm they will be operating in and fully appreciate the nature of the chosen paradigm.

The underlying research philosophy for this research is pragmatism. Research in pragmatism begins with a problem and attempts to give practical answers which may impact future practice (Saunders et al., 2007). This research aimed to identify and establish a hierarchy of the critical success factors that affect Agile software development success as perceived by Agile practitioners in the South African software

development industry. The findings were then formulated into a framework that other agile practitioners in

3.4 RESEARCH APPROACH

The second layer of the 'research onion' shows the different forms of reasoning used in theory development - abduction, induction, and deduction. A topic with a great deal of information in one context but significantly less in the context of your study may lend itself to an abductive method, allowing you to alter an existing theory (Saunders et al., 2007).

As stated in the problem statement, there Is not much research on this topic in a South African context. Additionally, some inconsistencies have been identified in existing literature that suggest a need for further research. Thus, abductive reasoning was deemed the most suitable. It is used to make sense of unexpected or confusing facts to fill gaps in our ideas and preserve or restore their coherence (Żelechowska et al., 2020). Furthermore, abduction is versatile and may be employed by researchers from various research philosophies (Saunders et al., 2007).

3.5 RESEARCH STRATEGY

The third layer of the 'research onion' depicts methodological choice, and the fourth layer depicts the strategies. Research using pragmatism uses various methods and strategies such as mixed, multiple, qualitative, quantitative, and action research, emphasising practical results (Saunders et al., 2007). Consequently, this research used a qualitative approach and a case study strategy.

Yin (1994) defines a case study as "an empirical inquiry that investigates a contemporary phenomenon within its real-life context". The case study method is a common form of qualitative analysis as it studies depth instead of breadth (Goddard and Melville, 2004). According to Hesse-Biber and Leavy (2011), there is distinctiveness in using the case study, which allows researchers to get comprehensive knowledge of a phenomenon in its context. The case study was a cross-sectional

single case study. Single case studies allow researchers to examine phenomena extensively and get an in-depth explanation and understanding (Darke et al., 1998). Thus, a case study is an ideal strategy for a complex phenomenon that requires in-depth holistic inquiry (Dubé and Paré, 2003).

The IS field has seen a turn from technological to organisational inquiries and, subsequently, more attention to context and innovations related to their context (Benbasat et al., 1987). Thus, case studies are now commonly used within IS (Shanks and Bekmamedova, 2018). Benbasat et al. (1987) identified three strengths of the case study strategy in IS: 1) developing theories from experience, 2) comprehending the nature and complexity of IS processes, and 3) acquiring significant insight into new topics. These strengths combine to make a strong case for using a case study to address the research problem since it is complex in nature, aims toward theory development from practitioners' experiences, and is a relatively new topic. This aligns with the ontology of pragmatism which Saunders et al. (2007) says is one were an external, complex, and rich 'Reality' is the practical outcome of ideas, experiences, and practices.

Given the nature of the research question and the pragmatism position, a case study is an appropriate strategy. It is ideal for capturing software practitioners' knowledge and theory development (Benbasat et al., 1987). The case study strategy provided a holistic, in-depth understanding of the critical success factors of Agile software development in the South African software industry by studying a single organisation and eliciting knowledge from Agile practitioners.

In the case study, there is a concept called "unit of analysis”, and it defines what the case study focuses on, such as a group, an organisation and an individual (Grünbaum, 2007). When the unit of analysis is individuals, the aim is to get the interpretations of social actors' perceptions of a particular phenomenon or meanings that actors ascribe to a phenomenon (Grünbaum, 2007). Thus, individuals are poised as a suitable unit of analysis as it will aid in achieving the primary research objective: the aim of the study to identify and establish a hierarchy of the critical success factors that affect Agile

software development success as perceived by Agile practitioners in the South African software development industry.

3.6 RESEARCH METHOD

The two main types of research methods are quantitative and qualitative (Creswell, 2009). Researchers usually select the quantitative method to answer research problems needing numerical data and the qualitative method for research problems needing textural data (Williams, 2007). In other words, quantitative approaches collect data through statistics and figures, while qualitative approaches gather data through words, images, or objects (Peters, 2020).

For this study, the qualitative research method was selected. Qualitative research explores and gains insights into how people understand a situation or problem (Creswell, 2009). It gives the researcher holistic insights into deep, contextual data by allowing them to engage in conversation with research participants in a natural setting (Jick, 1979, Creswell and Creswell, 2017, Mason and May, 2002). By doing qualitative research, the researcher interacted with the Agile practitioners in the South African software development industry to elicit a richer understanding of the critical success factors in Agile software development.

It is noted that the dominant research method in the subject area of Agile success factors is quantitative, as discussed in the research problem section. However, according to Hummel and Epp (2015), this focus on the quantitative assessment of Agile software development success factors that earlier studies had resulted in somewhat contradicting results (Hummel and Epp, 2015). Through a qualitative approach, this research aims to add depth to the study of the phenomena and potentially contribute to resolving the partial contradictions produced by quantitative research.

3.7 DATA COLLECTION

One of the advantages of the case study strategy is its adaptability and flexibility, which lets a researcher use single or multiple techniques for data collection to explore a research problem (Cavaye, 1996). Researchers who use the qualitative method typically gather information from various sources and contexts, increasing the robustness and validity of results (Kaplan and Maxwell, 2005). For this research, the sources of data were in-depth, semi-structured interviews of individuals with different roles in a software development project. The choice of data collection sources and methods was motivated by the need to gain a deeper understanding of the research topic and increase the study's accuracy.

One of qualitative research's most common and vital data collection tools is interviews (Myers and Newman, 2007). Qualitative interviews in research act as night-vision goggles (Myers, 2019), allowing us to see what is not typically visible and to examine what is usually looked at but hardly ever seen (Rubin and Rubin, 2011). The interviews were conducted with Agile practitioners, including software developers, user experience (UX) designers, user interface (UI) designers, UX analysts, project managers, functional managers, technical leads, testers, and business analysts.

Using multiple data sources in the same study is one way of triangulation (Hussein, 2009). Triangulation in research is "the use of more than one approach to researching a question" (Heale & Forbes, 2013). In research, triangulation can refer to utilising two or more theories, data sources, methods, or researchers. (Heale and Forbes, 2013). Data source triangulation was used in this research. Kaplan and Maxwell (2005) define data triangulation as the process of obtaining and analysing data from multiple sources to get a holistic picture of the research topic. Three data triangulation techniques exist: time, space, and person (Hussein, 2009). Hence, the interviews in this study used people with different roles and years of experience to have triangulation on persons. Variation persons add to the study since they can disclose unusual data or detect comparable patterns, enhancing confidence in the conclusions. (Fielding et al., 1986)

The participants were informed of the leading research goal without giving too much detail that could bias their opinions on the research issue. The interviews were held over two months, from March 2023 to April 2023. The interviews were recorded and transcribed using Otter and coded using Atlas.ti, a qualitative research analysis tool.

Before the final interview structure was set, a pre-test interview was conducted. The interview structure for the pre-test was informed by the theoretical framework developed in section 2.6. The first section of the question was to elicit if the participants were eligible to participate in the study. The second section had questions designed to get the interviewee's background information that was presented as evidence of triangulation. The last set of questions was to identify the critical success factors of agile software development. The categories of CSF identified in the systematic literature review were used to develop the questions. However, there was an additional question where the participants could list any CSF they did not consider falling into any category.

The Pre-testing is done to identify and address issues with the research instrument before a set of procedures is finalised (Wolf et al., 2016). Many authors deem pre-testing a research instrument necessary (Collins, 2003, Presser et al., 2004, Wolf et al., 2016). Backstrom and Hursh (1963), "No amount of intellectual exercise can substitute for testing an instrument designed to communicate with ordinary people". The goal is to ensure that the interview participants understand the question concepts; they do so consistently and in the way that the researcher intended (Collins, 2003). Thus, for this research, a pre-test was conducted on the questions before conducting the interviews.

Several pre-testing methods include cognitive interviewing, usability testing/eye tracking, Interviewer debriefing, behaviour coding, psychometric/Item response theory analysis, planned experiments and field-based probing (Wolf et al., 2016). In this research, cognitive interviewing was done with four agile software practitioners. In addition, two academics mailed in their reports on the questionnaire. Their academic research focuses on agile software development and thus would provide expert input. The four agile practitioners represented the homogenous population with varying roles,

years of experience, and age. All the pre-test interviews were recorded using Otter and analysed using Atlas.ti. The procedure followed for the pretest, the demographics of the pretest participants and the findings are detailed in Appendix D.

Pretesting assisted the researcher in testing the research tool, and the findings were used to modify the interview questions that were unclear to participants. Appendix E shows the final interview structure produced and used in the case study. Before conducting the interviews, the researcher used secondary data from the case study organisation's website to learn more about the company, and the owners provided details on the company structure as this was not on the website. This information assisted in providing a holistic understanding of the company.

3.8 SAMPLING

Sampling in qualitative research aims to gather particular cases, events, or behaviours that can clarify or enhance the researcher's comprehension of the studied topic (Ishak and Abu Bakar, 2014). The sampling method used in this study aimed to identify analytical units that might enhance the existing research on critical success factors and bring new knowledge on their level of importance to each other. For that reason, non-probability sampling methods were used. Non-probability sampling is a sampling technique that chooses a set of participants for a study using non-random methods (Etikan and Bala, 2017).

3.8.1 Target population

The population of interest is Agile practitioners in South Africa. The case study strategy was chosen for this research; thus, the sample frame is Agile practitioners at the selected South African software organisation.

3.8.2 Sampling method

Two non-probability sampling methods were used in this research. Convenience sampling was used to select the organisation for the case study, while purposive sampling was used to select the participants for the interviews within the organisation.

Convenience sampling selects the case to be studied based on how easily the researcher can reach it (Kulathunga and Ratiyala, 2018). Thus, the selected organisation for the case study met the following practical criteria listed by Dörnyei (2007): 1) easy accessibility, 2) geographical proximity, 3) availability at a given time, and 4) willingness of organisational members to participate.

Purposive sampling is a non-probability sampling method applied to choose a sample of individuals from a population (Etikan and Bala, 2017). It allows the researcher to apply judgment when choosing participants (Peters, 2020). The researcher wanted to ensure that an Agile practitioner with the highest number of years of experience in Agile and the lowest years of experience represented all the roles in the case study organisation. The organisation's management provided this information. Thus, the flowing roles were represented by at least two individuals: software developer, UX designer and UI Designer. However, in the organisation, the following roles only had one participant who was available and willing to participate: project manager, functional manager, technical lead, business analyst and tester. The researcher also applied their judgement and thought it might be beneficial to diversify the pool as much as possible. Hence, the other participants invited were individuals who worked remotely and in varying team sizes.

In qualitative research, it is necessary to represent various voices by interviewing diverse people within an organisation (Myers & Newman, 2007). This is called triangulation for participants (Rubin & Rubin, 2011). Myers and Newman (2007) argue that to overcome several biases, such as elite bias, researchers must involve various participants in their sample at different organisational levels. Elite bias is the overweighting of data from well-informed, articulate, typically high-status informants and underrepresenting data from less articulated, lower-status informants (Miles & Huberman, 1994). With this method, sample members are chosen based on their expertise, knowledge, and associations concerning the research subject (Langkos, 2014)

Qualitative research that emphasises in-depth investigation in a small group normally uses purposive sampling rather than random sampling since importance is placed on quality rather than quantity (Bowen, 2005). It allows the researcher to gain a deeper insight into the problem from the most suitable participants (Asnawi, 2012). Thus, purposive sampling was used to select the interview participants.

3.8.3 Sample size

Qualitative researchers are not worried about sample size and rarely get a large sample from the studied population (Ishak and Abu Bakar, 2014). However, "many qualitative researchers have been criticised for not explaining their sample size decisions in their research" (Boddy, 2016:1). Therefore, it is important to discuss the sample size for the interview participants in this research, which was influenced by theoretical saturation.

Theoretical saturation is useful when designing qualitative research (Boddy, 2016). Data collection is saturated when no new elements are found in interviews. Adding new data is no longer required because doing so will not alter how the researcher understands the phenomenon being examined. (Nascimento et al., 2018). Rowlands et al. (2016) found that 95% theoretical saturation confidence was reached in 11 and 13 interviews. Boddy (2016) illustrated that data saturation usually occurs at 12 interviews within a somewhat homogeneous population. The researcher used this estimated initial number of 12 participants. Thus, 12 interviews were initially conducted. After the 12 interviews, the researcher conducted the thematic analysis. A few new labels (codes) were added in the 12th interview; thus, the interview was scheduled one more at a time and conducted the thematic analysis. At 15 interviews, no new labels (codes) were identified in the last two interviews. Thus, the researcher determined that theoretical saturation was reached and ended at 15 interviews. When theoretical constructions match with current data and comparison of theoretical constructs with fresh data offers no substantial new insights, the point of diminishing returns is reached (Gasson, 2004).

3.9 DATA ANALYSIS

The main objective of qualitative data analysis is to comprehend the pursuit of order and coherence (Kaplan and Maxwell, 2005). Qualitative research is iterative; therefore, throughout the coding process, the researcher can modify codes and themes that appear to fit the concepts better (Braun and Clarke, 2006). There are four fundamental methods of qualitative analysis detailed by (Kaplan and Maxwell, 2005:41), namely: "(1) coding, (2) analytical memos, (3) displays, and (4) contextual and narrative analysis".

This research used coding, specifically thematic analysis since the research took an abductive approach. One of qualitative research's most popular data analysis methods is thematic analysis. (Majumdar, 2022). A thematic analysis is one where themes, categories, and analysis patterns are derived from the data instead of being applied to precede the data collection and analysis (Bowen, 2005). This type of analysis explains research data in an organised way (Boyatzis, 1998).

Qualitative methods yield large volumes of data that might not be immediately flexible for analysis, so computer software can be used to facilitate the analysis (Kaplan and Maxwell, 2005). The interviews were recorded and transcribed using Otter, a text transcription tool. Additionally, the coding was done using Atlas.ti, qualitative data analysis, and research software. The transcriptions were uploaded onto Atlas.ti, and the researcher read through each interview and generated terms or phrases to label a section of text. Once a label was created, it could be reused. Once all the relevant text in the interviews was labelled, an extra step was taken to consolidate the list of codes. During this step, the coded data is reviewed to discover areas of resemblance and overlap between codes (Braun and Clarke, 2012). If a code had a partially similar meaning to another code in such a way that it could be used to describe that code, it was labelled as an attribute of that code. This process led to the identification of the critical success factors. The critical success factors were then ranked using frequency-based ranking.

According to Lincoln and Guba (1985), qualitative research researchers must consider trustworthiness. The confidence level in data gathered, interpretation, and methods employed to ensure research quality is called trustworthiness (Polit and Beck, 2014). Lincoln and Guba (1985) state that the criteria to consider when establishing trustworthiness are: transferability, credibility, confirmability, and dependability are the four aspects. A fifth aspect was added in 1994 called authenticity (Elo et al., 2014).

Credibility is confidence in the truthfulness of the results (Bowen, 2005). This study uses multiple data sources (data triangulation) discussed in the data collection section to establish credibility. Through data triangulation, the scholar tries to provide a convergence of evidence that leads to credibility (Eisner, 2017).

Confirmability is the level at which the results are consistent and repeatable (Connelly, 2016). Methods to achieve confirmability include maintaining an audit trail of analysis (Lincoln and Guba, 1985) or methodological memos (Polit and Beck, 2014). This study had thorough notes of all decisions made during the analysis, which helped synthesise the identified codes into factors.

Dependability measures the consistency of the results over time (Bowen, 2005). It depends on the data analysis and collection accuracy (Lincoln and Guba, 1985). Measures for dependability include peer debriefings and maintaining an audit trail (Connelly, 2016). This aspect was handled in the same way as conformability.

"Transferability" refers to other researchers' ability to apply a study's findings to their research setting (Bowen, 2003). The researcher ensured transparency throughout the research and gave findings with "dense" descriptions of the phenomena to ensure transferability. A vivid picture of the research analysis resonates with readers and makes the research more useful to people in different settings (Amankwaa, 2016).

The degree to which researchers impartially depict a variety of distinct realities and accurately represent the lives of the participants is called authenticity. (Polit and Beck, 2014). This aspect was addressed through the careful selection of interview participants, consideration of theoretical saturation and the extensive detailing of the

participants' responses. Selecting acceptable people for the study sample and providing rich, comprehensive descriptions aids in gaining authenticity (Connelly, 2016)

3.10 ETHICS

The research ethics approval was granted by the Ethics Committee of the Faculty of Engineering, Built Environment and IT, University of Pretoria (Appendix F). The case study company and interviewee participants were sent an invitation letter, shown in Appendix H. The case study organisation mailed a signed letter of authorisation, shown in Appendix G, and provided consent to participate in the research. Before the interviews commenced, all participants were asked to consent to participate in the research by signing a consent letter shown in Appendix I—the letter aimed to reassure participants that their participation in the study was voluntary. Participants were free to withdraw from the study at any point for any reason. Participants were told about the study's objectives and reassured that their anonymous responses would only be used for this research. Lastly, the researcher created a comfortable setting for all participants.

3.11 CHAPTER SUMMARY

This chapter detailed the research design, data collection methods, sampling methods, data analysis process, and research ethics considerations. A summary of this chapter is shown in Table 9, showing the key decisions made to conduct this study.

Table 3.1: Key decisions made to conduct this study.

Level of decision	Choice
Research Philosophy	Pragmatism
Research Approach	Abductive
Research Strategy	Single case study
Research Method	Qualitative
Research Techniques	Semi-structured Interviews
Sampling Method	Convenient Sampling – Case study selection Purposive Sampling – Interview Participants
Sample size	15 interviews
Data Analysis	Thematic Analysis

4 RESULTS

4.1 INTRODUCTION

This chapter presents the data collected through semi-structured interviews at a software development organisation in South Africa. The first section of the chapter presents an analysis of the case study organisation that participated in the case study. The second section presents the descriptive statistics of the interview participants. This section is vital as it helps with understanding and interpreting the data collected from the study participants. The following section describes how the collected data was consolidated to produce the critical success factors. Then, the last section presents the consolidated list of the critical success factors identified from the semi-structured interviews ranked by the frequency of mentions of the factor and its attributes.

4.2 CASE BACKGROUND

The case study was conducted on a South African software development company. The company and interviewee names will not be used for reasons of confidentiality. The company will be referred to as the case study organisation in the rest of the research, and interviewees will be referred to by numbers from 1 to 15.

The case study organisation is a human-centred software organisation that focuses on developing custom-designed systems using Agile software development practices. The case study organisation has been operating for eight years and has 39 employees. There are two main departments, which they refer to as teams. These teams are the software development, commonly called “the dev team”, and the design and analysis, commonly called “the D&A team”. The dev team focuses on front-end website development, back-end website development and application engineering. This includes custom systems development and system integration. The D&A team has focused on user interface (UI) design, user experience (UX) design and business and systems analysis. This includes wireframing, interaction design, prototyping, user testing, usability testing, UX research and strategy development.

The case study organisation has over 30 projects across about 20 clients. All the projects are in different industries, including tourism, banking, medical aid, insurance, automation, telecommunications, and information technology. They use different agile

methodologies for different projects based on the type of software solution required and the client's requirements. However, the most frequently used methodology is SCRUM.

The owners referred to the structure of their organisation as relatively flat. There are two owners/founders who also act as the organisation's managers. In addition, there is one project manager, two leads for the dev team, and two leads for the D&A team, who all make up the organisation's management committee. The rest of the employees are team members of either the dev team or the design team, except for the administration staff, who are not involved in the software development projects.

4.3 DESCRIPTIVE STATISTICS OF INTERVIEWEE DEMOGRAPHICS

In the interview, screening questions were asked to ensure that the participants were eligible to participate. These questions were about their age, years of experience with Agile and the number of Agile projects they had been on. The requirements were that the participants needed to be over 18, have at least three months of experience, and have been on at least a part of one project.

Secondly, background questions were asked regarding the interviewees' job titles, work environment, recently used Agile methodology, and team size to ensure the participants represented a range of roles and years of experience. All participants met the criteria and were eligible to participate. Thus, 100% of the participants were over 18. The rest of the data from questions 1.2 to 3.5 was analysed using descriptive statics and detailed below.

4.3.1 Question 1.2: Years of experience

Table 10 shows the distribution of the interviewees' years of experience. 40% of the participants had one to four years of experience, and similarly, 40% of the participants had over four years of experience. 13.3% of the participants had three to four years of experience, and only one participant had between three to twelve months of experience. The remaining 13.3% of participants had three to four years of experience.

Commented [DH7]: Rephrase - doesn't make sense?

Table 4.1: Descriptive statistics of interviewees' years of experience

Measure	Frequency	%
Years experience		
3 - 12 months	1	6,7
1 - 2 years	6	40,0
3 - 4 years	2	13,3
4+ years	6	40,0
Total	15	100

4.3.2 Question 1.3: The number of projects

Table 11 shows the descriptive statistics of the interviewees' number of projects. Most of the participants had been a part of over ten projects, making up 53.3% of the group. 26.7% have been part of six to ten projects, with the remaining 20% of the participants having taken part in one to five projects.

Table 4.2: Descriptive statistics of interviewees' number of projects

Measure	Frequency	Percentage (%)
No of projects		
1 - 5	3	20,0
6 - 10	4	26,7
10+	8	53,3
Total	15	100

4.3.3 Question 2.1: Job title

Table 12 shows the descriptive statistics of the interviewee's job titles. 33.3% of the participants were software developers. UX designers and UI designers each were 13.3%. The rest of the participant's job titles were distributed evenly: UX analyst - 6.7%, project manager – 6.7%, functional manager 6.7%, technical lead – 6.7%, tester – 6.7% and business analyst – 6.7%

Table 4.3: Descriptive statistics of interviewee job titles

Measure	Frequency	Percentage (%)
Job title		
Software Developer	4	33.3
UX Designer	2	13.3
UX Analyst	1	6.7
UI Designer	2	13.3
Project Manager	1	6.7
Functional Manager	1	6.7
Technical Lead	1	6.7
Business Analyst	1	6.
Tester	1	6.7
Total	15	100

4.3.4 Question 2.2: Working environment.

Table 13 shows descriptive statistics in an interviewee's working environment. 60% of the participants work hybrid, meaning they come into the office a few days a week and work some days remotely. 26.7% of participants work from the office every day of the week. 13.3% of participants work remotely every day of the week.

Table 4.4: Descriptive statistics in an interviewees' working environment.

Measure	Frequency	Percentage (%)
Work environment		
Hybrid	9	60,0
Office	4	26,7
Remote	2	13,3
Total	15	100

4.3.5 Question 2.3: Agile methodology used in the most recent project.

Most of the participants used Scrum in their most recent project, making up 73% of the participants, as shown in Table 14. 3% of the participants said they used a hybrid of SCRUM and Kanban using different elements from both. 7% of participants, which is one participant, used Kanban.

Table 4.5: Descriptive statistics of interviewees' most recent agile methodology

Measure	Frequency	Percentage (%)
Recently used Agile methodology.		
A hybrid of Scrum and Kanban	3	20
Scrum	11	73
Kanban	1	7
Total	15	100

4.3.6 Question 2.4: Team size on your most recent project?

Table 15 shows the descriptive statistics of the interviewees' team size in their most recent project. The most common range for a team was one with four to six team members, represented by 46.7%. Secondly, 26.7% of the interviewees were in a team with one to three members. 20% of the interviewees were in a team with seven to ten members. Lastly, 6.7% were in a team of more than ten members.

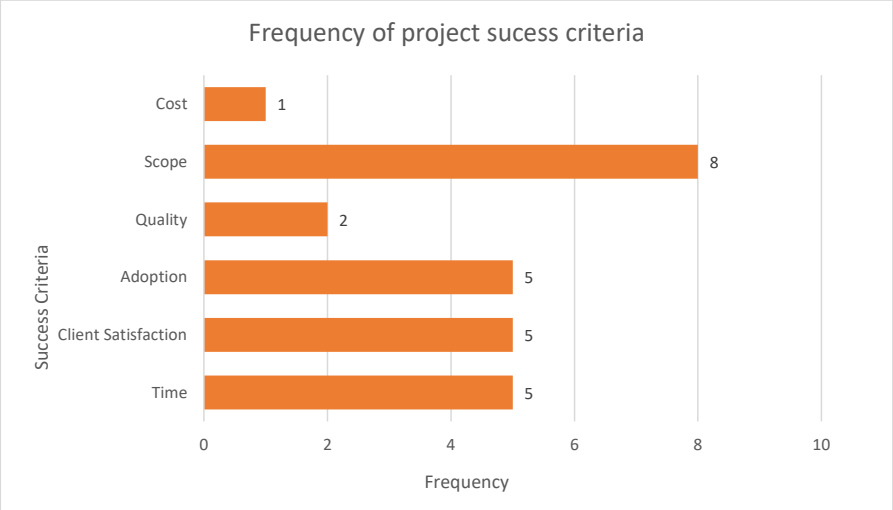
Table 4.6: Descriptive statistics of interviewees' team size

Measure	Frequency	Percentage (%)
Team size		
1 - 3	4	26,7
4 - 6	7	46,7
7 - 10	3	20,0
10+	1	6,7
Total	15	100

4.3.7 Question 2.5: Criteria for Project Success

Furthermore, participants were asked what criteria needed to be met for a project to be deemed a success from their point of view. Six criteria for agile software development were identified: cost, scope, quality, adoption, client satisfaction and time. The respondent's data helped assess whether the success criteria in the initial theoretical framework would need to be amended. Figure 9 below shows the frequency of the perceived success criteria.

Figure 4.1: Frequency of project success criteria



4.4 DESCRIPTIVE STATISTICS OF IDENTIFIED CODES

In the core questions of the interviews, the interviewees were asked questions that would reveal the factors they believe to be critical to the success of an agile software development project. There was a question for each category identified in the systematic literature review: organisational, team, customer, process, technical, and project. In the last question, they were asked to add any other factors without a category, and this was to give room to identify any other categories or factors that they considered to not fit into any category according to our definition or their understanding.

The responses were analysed through thematic analysis using Atlas.ti. It is a program that assists in analysing qualitative data for qualitative research. The software was used to code the interview transcripts. When analysing the data, if a snippet of the interview described a factor, it was highlighted and given a label called a code. The codes were then used to label other similar texts in the interviews. The complete list of initial codes is listed in Appendix J. In this section, they are discussed per category.

The categories used were those identified in the systematic literature review. This section will refer to the categories as code groups. At least one code was identified in

each interview, with some interviewees mentioning more than others. Table 16 below shows the number of codes identified per category in each interview.

Table 4.7: Number of codes identified per code group in each interview.

Interview ID	Organisational	Customer	Team	Technical	Process	Project
1	10	6	10	2	4	5
2	14	2	9	3	5	1
3	6	9	8	3	5	3
4	9	6	4	3	6	1
5	14	4	15	1	8	1
6	3	3	5	3	6	0
7	2	4	4	2	10	2
8	2	0	3	2	5	0
9	10	3	11	2	9	2
10	4	4	3	1	1	2
11	3	0	2	1	4	2
12	2	4	9	1	2	5
13	2	2	8	1	3	1
14	4	1	5	1	3	1
15	12	3	8	0	8	2
Average	6	3	7	2	5	2
Minimum	2	0	2	0	1	0
Maximum	14	9	15	3	10	5

The average number of codes identified in the organisational code group was six, the minimum was two, and the maximum was fourteen. The average number of customer codes mentioned in an interview was three. The maximum number of customer codes mentioned by a single interviewee was nine, and the minimum number of mentions was zero. For team codes, the average was seven, with the minimum mentions being two and the maximum fifteen. Process codes had an average mention of five codes per interview, with a minimum of one and a maximum of ten. Technical codes had the lowest numbers, with the average number of mentions being two, a maximum of three and a minimum of zero. Lastly, the average number of mentions for project codes was two, with a maximum of five and a minimum of zero. When asked for additional codes in question 3.7, some interviewees had nothing to add. In contrast, others added more codes relating to previous questions, which are included in the table above in the

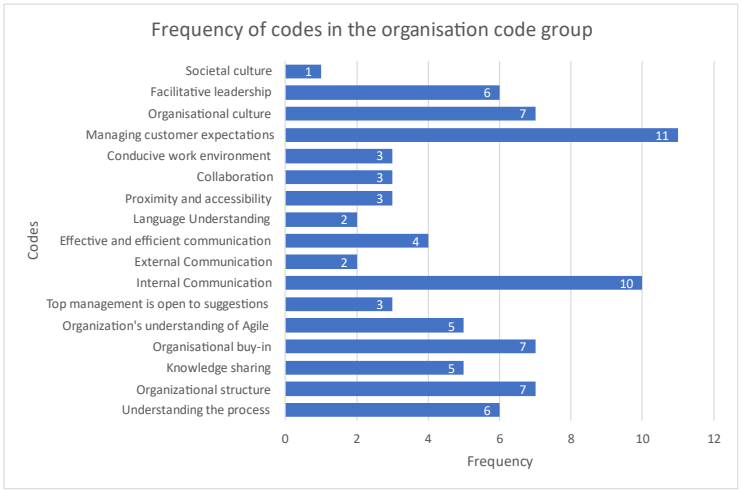
relevant category. The sections below will detail the frequency of mentions of each code.

Question 3.1: Organisational

In your experience, what organisational factors (factors relating to the organisational structure and administrative climate of the company) affect the success of an agile software development project?

In the interviewees' responses, 17 codes were identified, as shown in Figure 10. The code identified in all but one interview was managing customer expectations, with 11 mentions, followed by Internal communication, with 10 mentions. The code with the lowest mentions was a societal culture with one mention. The rest of the codes were distributed as follows: organisational buy-in (7), organisational structure (7), facilitative leadership (6), understanding the process (6), knowledge sharing (5), organisation understanding of agile (5), effective and efficient communication (4), top management open to suggestions(3), collaboration (3), conducive work environment (3), proximity and accessibility (3), language understanding (2) and external communication (2).

Figure 4.2: Frequency of codes in the organisation code group

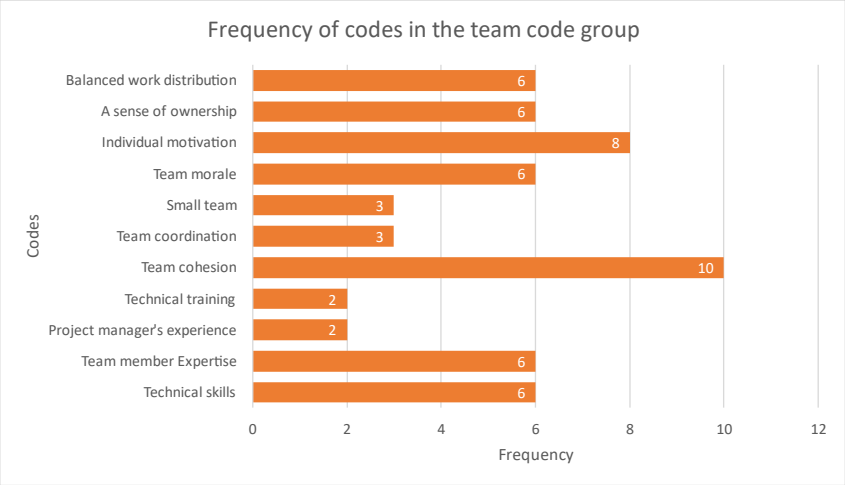


4.4.1 Question 3.2: Team

In your experience, what team factors (factors relating to the people who manage and execute the project) affect the success of an agile software development project?

The team code group had 11 codes initially. The codes are shown in Figure 11, and as can be seen, the factor with the most mentions was Team cohesion, with ten mentions. The second highest was individual motivation, with eight mentions. The rest of the codes were distributed as follows: understanding project requirements (8), individual motivation (6), team member expertise (6), a sense of ownership (6), team morale (6), balanced work distribution (6), technical skills (6), team coordination (3), small team (3), technical training (2), project manager's experience (2).

Figure 4.3: Frequency of codes in the team code group

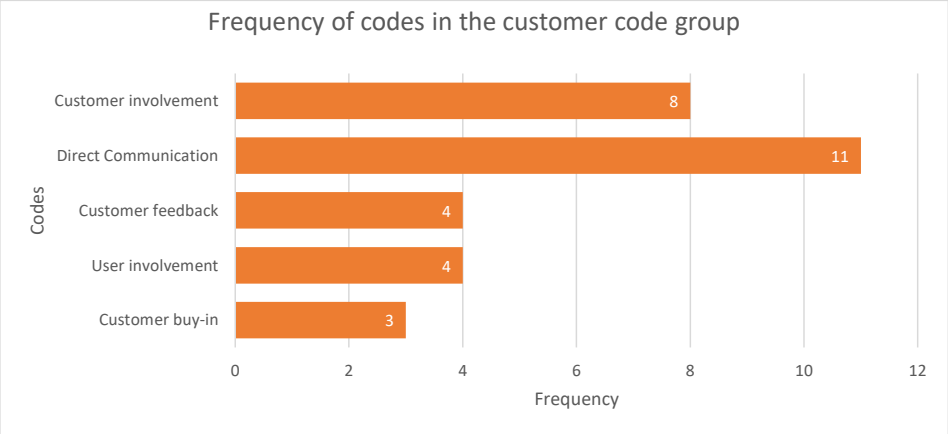


4.4.2 Question 3.2: Customer

In your experience, what customer factors (factors relating to the people who sponsor the project or will use the product) affect the success of an agile software development project?

Five codes were identified in the interview responses, as shown in Figure 12. The code mentioned the most was direct communication, with 11 mentions. The rest of the codes mentioned were as follows: customer involvement (8), user involvement (4), customer feedback (4), and customer buy-in had three mentions.

Figure 4.4: Frequency of codes in the customer code group



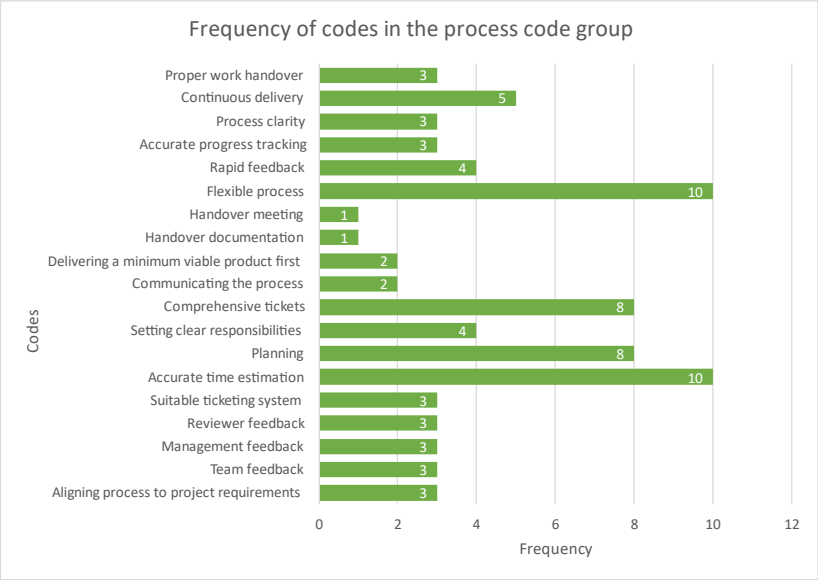
4.4.3 Question 3.4: Process

In your experience, what process factors (factors relating to how project activities are carried out) affect the success of an agile software development project?

There were 19 process codes identified in the interview analysis, as shown in Figure 13. The codes with the highest mentions were accurate time estimation and flexible process with ten mentions. The rest of the codes were distributed as follows: continuous delivery (5), rapid feedback (4), setting clear responsibilities (4), proper work handover (3), process clarity (3), accurate progress tracking (3), suitable ticketing system (3), reviewer feedback (3), management feedback (3), team

feedback (3), aligning process to project requirements (3), delivering a minimum viable product (2), communicating the process (2) and with the least amount of mentions handover meeting (1) and handover documentation (1).

Figure 4.5: Frequency of codes in the process code group

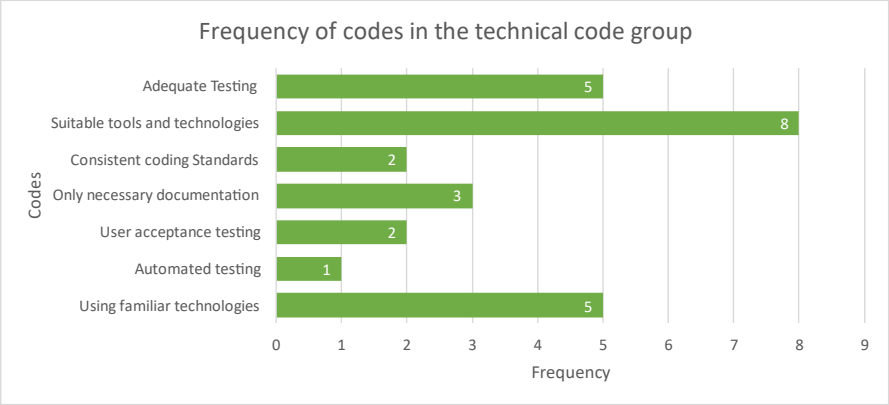


4.4.4 Question 3.5: Technical

In your experience, what technical factors (factors relating to the tools, technologies, or techniques used in the project) affect the success of an agile software development project?

The question on technical codes had seven identified codes, as shown in Figure 14. Having suitable tools and technologies had the most mentions as all eight interviews mentioned this code. Secondly, using familiar technologies and adequate testing with five mentions. Only necessary documentation had three mentions; consistent coding standards and user acceptance testing had one mention, as did automated testing.

Figure 4.6: Frequency of codes in the technical code group

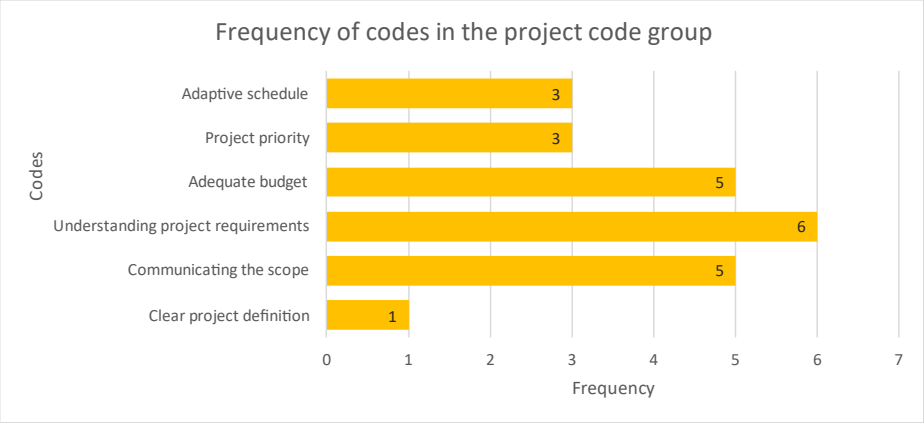


4.4.5 Question 3.5: Project

In your experience, what project factors (factors relating to the project parameters) affect the success of an agile software development project?

Six project-related codes were identified, as shown in Figure 14. Understanding project requirements had the highest number of mentions, with six. Second was adequate budget and communicating the scope, with five mentions. Adaptive schedule and project priority both had three mentions. Lastly, a clear project definition had one mention.

Figure 4.7: Frequency of codes in the project code group



4.5 SYNTHESIS OF CRITICAL SUCCESS FACTORS

From the thematic analysis of the interviews, 59 codes were initially identified, and six criteria were identified as the criteria the interviewees considered to define project success. The research by Chow and Cao (2008) and (Stankovic et al., 2013) provided attributes for the critical success factors they identified. It is stated that they used Cronbach's coefficient alpha to narrow down the factors they initially identified from their surveys to the critical success factors since their research was qualitative and exploratory. However, how they identified the attributes is not clearly stated. In this research, the attributes were identified from the initial list of codes and the interviewee's descriptions of the factors.

The researcher identified and coded all data relevant to the research question during the initial coding, as presented in Section 4.4. However, an extra step was taken to consolidate the list of codes. During this step, the coded data is reviewed to discover areas of resemblance and overlap between codes (Braun and Clarke, 2012). If a code had a partially similar meaning to another code in such a way that it could be used to describe that code, it was labelled as an attribute of that code.

For example, Agile software engineering techniques is a CSF in existing literature (Agarwal et al., 2018, Chiyangwa, 2017, Chow and Cao, 2008, Stankovic et al., 2013, Stelzmann et al., 2010, Wagener, 2012). Chow and Cao (2008) described this factor using five attributes: well-defined coding standards upfront, a suitable amount of documentation, seeking simple design, extensive refactoring activities and accurate integration testing. Two of these attributes, only necessary documentation and consistent coding standards, were identified in the interviews as factors and thus were grouped into the factor “Agile software development techniques.”

Some factors were combined to form one factor because they are commonly associated with literature and were often mentioned by the interviewees. The shared mentions were counted as one. For example, two factors, “team cohesion” and self-organising team,” were merged into the factor “coherent-self organising team”. The same was done for planning and accurate time estimation.

Some factors were grouped under an all-encompassing term where they did not describe each other but had descriptions used in literature to describe another factor. For example, in the SLR, the following attributes are described: Team capability: technical solid skills knowledge, knowledge about the system, knowledge about the methods, high capabilities, explicit domain knowledge and real-world experience in the technology domain. Hence, the following codes: technical skills, team member expertise, project manager's experience and technical training were grouped under “Team capabilities”.

Table 17 shows the synthesised list of critical success factors identified as attributes and the related CSF.

Table 4.8: Synthesised list of identified critical success factors attributes.

Critical success factor	CSF Freq	Attribute	Attribute Freq	Combined Freq	Category
Customer Involvement	8	Customer buy-in	3	30	Customer
		User involvement	4		
		Customer feedback	4		
		Direct Communication	11		
Managing customer expectations	11	Understanding the process	6	17	Organisational
Organisational culture	7	Organisational structure	7	31	Organisational
		Knowledge sharing	5		
		Organisational buy-in	7		
		Organisation's understanding of Agile	5		
Facilitative leadership	6	Top management is open to suggestions	3	9	Organisational
Communication	-	Internal Communication	10	16	Organisational
		External Communication	2		
		Effective and efficient communication	4		
Societal culture	1	Language Understanding	2	3	Organisational
Conducive, collaborative work environment	-	Proximity and accessibility	3	9	Organisational
		Collaboration	3		
		Conducive work environment	3		
Flexible process	10	Aligning process to project requirements	3	16	Process
		Team feedback	3		
Rapid feedback	4	Management feedback	3	10	Process
		Reviewer feedback	3		
Accurate progress tracking	3	Suitable ticketing system	3	6	Process
Accurate planning and estimation	-	Accurate time estimation	10	30	Process
		Planning	8		
		Setting clear responsibilities	4		
		Comprehensive tickets	8		
Process clarity	3	Communicating the process	2	5	Process
Continuous delivery	5	Delivering a minimum viable product first	2	7	Process
Proper work handover	3	Handover documentation	1	5	Process
		Handover meeting	1		
Clear scope and requirements	-	Clear project definition	1	12	Project
		Communicating the scope	5		
		Understanding project requirements	6		
Adequate budget	5	-	5	5	Project
Project priority	3	-	3	3	Project

Critical success factor	CSF Freq	Attribute	Attribute Freq	Combined Freq	Category
Adaptive schedule	3	-	3	3	Project
Team capabilities	-	Technical skills	6	16	Team
		Team member Expertise	6		
		Project manager's experience	2		
		Technical training	2		
Coherent – self-organising team	13	Team cohesion	-	13	Team
		Team coordination	-		
Small team	3	-	3	3	Team
Team morale	6	Individual motivation	8	26	Team
		A sense of ownership	6		
		Balanced work distribution	6		
Suitable tools and technologies	8	Using familiar technologies	5	13	Technical
Adequate Testing	5	Automated testing	1	8	Technical
		User acceptance testing	2		
Agile software development techniques	5	Only necessary documentation	-	5	Technical
		Consistent coding Standards	-		

4.6 FREQUENCY-BASED RANKING OF THE CRITICAL SUCCESS FACTORS

A frequency-based ranking of the critical success factors was done and presented in this study. Since this was a qualitative study, the interviewees had the freedom to list as many Critical success factors as they could think of and describe them in their own words. In social representation, this is known as multiple response free association. Multiple response-free association entails asking an individual to produce several words or expressions without restriction relating to the study object (Dany et al., 2015).

The free association allows the researcher to process data produced directly from the free expression of individuals (Dany et al., 2015). One of the two rank indicators used in social representation is the frequency of an item and its associations, and the other is its appearance. This was used by some researchers in the IS field, such as Qatanani et al. (2021), Nguyen (2016), Belassi and Tukel (1996) and de Mello et al. (2019). The most cited factor(s) is ranked first, followed by the second most frequently mentioned factor, and so on (Belassi and Tukel, 1996). In their ranking, de Mello et al. (2019) associated each descriptive answer with the definitions in their catalogue, resulting in

evoked terms and associations. Similarly, the researcher identified this study's critical success factors (evoked terms) and their attributes (associations). Thus, the frequency ranking was done based on the number of mentions of a factor and its attributes.

Thus, the ranking shown in Table 18 presents prioritisation of the critical success factors based on the frequency of mentions by the agile practitioners in the study. In the case where the CSF had the same ranking, they were shown in alphabetical order.

Table 4.9: Frequency Based ranking of the critical success factors

Rank	Critical Success Factor	Total number of mentions in the interviewees (n – 301)		Category
		Combined Frequency	Percentage	
1	Organisational culture	31	10.3%	Organisational
2	Accurate planning and estimation	30	10.0%	Process
3	Customer Involvement	30	10.0%	Customer
4	Team morale	26	8.6%	Team
5	Managing customer expectations	17	5.6%	Organisational
6	Communication	16	5.3%	Organisational
7	Flexible process	16	5.3%	Process
8	Team capabilities	16	5.3%	Team
9	Coherent – self-organising team	13	4.3%	Team
10	Suitable tools and technologies	13	4.3%	Technical
11	Clear scope and requirements	12	4.0%	Project
12	Rapid feedback	10	3.3%	Process
13	Conducive, collaborative work environment	9	3.0%	Organisational
14	Facilitative leadership	9	3.0%	Organisational
15	Adequate Testing	8	2.7%	Technical
16	Continuous delivery	7	2.3%	Process
17	Accurate progress tracking	6	2.0%	Process
18	Adequate budget	5	1.7%	Project
19	Agile software development techniques	5	1.7%	Technical
20	Process clarity	5	1.7%	Process
21	Proper work handover	5	1.7%	Process
22	Adaptive schedule	3	1.0%	Project
23	Project priority	3	1.0%	Project
24	Small team	3	1.0%	Team
25	Societal culture	3	1.0%	Organisational

4.7 CHAPTER SUMMARY

This chapter presented the data collected in the semi-structured interviews conducted at the case study organisation. There were 15 interviews of different demographics that were described using descriptive statistics. The thematic analysis used the data analysis software Atlas.ti to analyse the qualitative data. The thematic analysis identified 65 codes of the critical success factors and their attributes. The codes were then synthesised to produce the 25 critical success factors. These 25 factors were then ranked based on the frequency of mentions to produce a list of the critical success factors.

5 DISCUSSION OF FINDINGS

5.1 INTRODUCTION

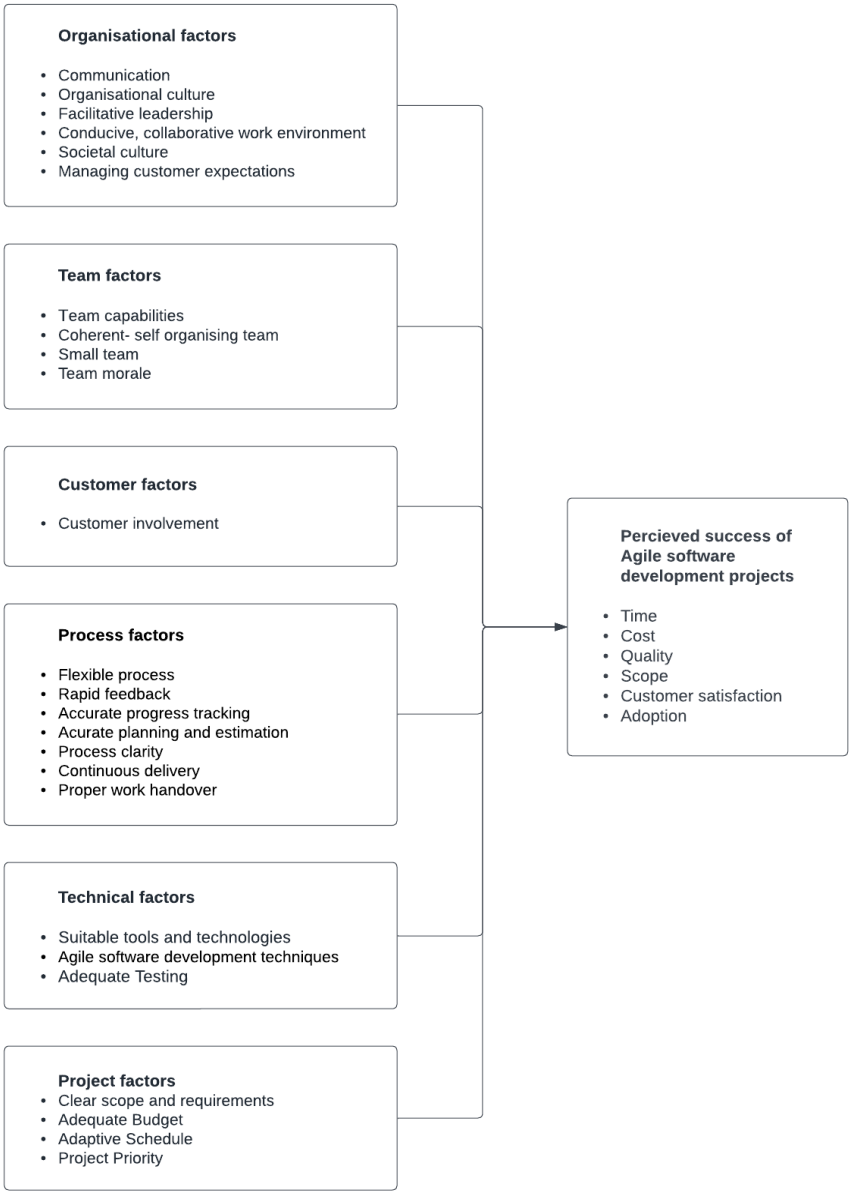
This chapter presents the results of the thematic analysis. Section 5.2 presents the framework developed from the data analysis and discusses the identified CSFs in their respective categories. This section is significant because it presents the factors that agile practitioners in a South African organisation deemed as critical success factors by answering the sub-research question a) *"What are the critical factors in the success of Agile software development recognized by Agile practitioners in the South African software development industry?".*

The following section, 5.3, presents the priority of the identified critical success factors. This section is significant because it answers the second research question. Sub-research question b) *What is the priority of the identified critical success factors?*

5.2 FRAMEWORK FOR THE CRITICAL SUCCESS FACTORS

A framework generated from the case study findings is shown in Figure 5-1. The framework displays the critical success factors of agile software development in the South African software development industry identified in the case study. Twenty-five factors were identified in organisational, team, customer, process, technical, and project factors. Each category factor is described below based on the descriptions mentioned by the interviewees in their respective categories. In addition, the most relevant quote to the interviewees was cited for each factor.

Figure 5.1: Framework for the CSF of agile software development in the South African software development industry



5.2.1 Agile software development project success

The interview findings identified six criteria for success in agile software development: time, cost, quality, scope schedule, customer satisfaction and adoption. The first five aspects were also identified in SLR and thus were in the initial framework. However, adoption was not identified in the initial framework. Thus, the " adoption " criterion was added to the definitions of success criteria. The following quotes from the interviews were used to influence the definition of adoption chosen from the literature:

"The biggest part of a project that shows success is adoption. It does not matter if you get out a product when no one is using it". (2)

"… we care about user experiences because that ultimately defines whether users will adopt the technology and continue using the product or service. And ultimately, that ends up being the success criteria. You can deliver something very quickly, like within two days, but if it is not excellent quality, and people do not want to use it, and they do not know how to use it, then you have not made that project a success, even if it was in time." (5)

"I think something usable at the end of the day should always be a success metric because you can develop things quickly, but if the software is unsuitable, that is pointless." (10)

Hence, the study defines adoption as the acceptance, integration, and use of a product (Salahshour Rad et al., 2018). Table 19 shows the success criteria and their definitions.

Table 5.1: Criteria for project success

Dimension	Element	Definition
Perceived level of success Agile software development projects	Time	Delivering on time
	Cost	Delivering within budget
	Scope	Meeting all requirements and objectives
	Quality	Delivering a good working product
	Customer Satisfaction	Customer approval of delivered product
	Adoption	acceptance, integration, and use of a product

5.2.2 Organisational Factors

The organisational factors category includes *factors relating to the organisational structure, organisational environment and administrative climate of the company* (Arcos-Medina and Mauricio, 2020, Chow and Cao, 2008, Hummel and Epp, 2015, Misra et al., 2006, Misra et al., 2009, Shakya and Shakya, 2020) The organisational factors identified in this study are managing customer expectations, communication, rapid feedback, organisational culture, facilitative leadership, conducive, collaborative work environment and societal culture. These factors are discussed below.

5.2.2.1 Managing customer expectations.

Managing customer expectations was not identified in existing literature and thus is poised as a new critical success factor or at least one that may particularly apply in the South African context. It was also ranked high in prioritisation in fifth place and the second highest-ranked organisational factor, highlighting its perceived significance.

The findings show that Agile practitioners believe an organisation needs to manage customer expectations and do so from the start of the project. *"You must start managing expectations early and planning accordingly as opposed to just jumping in and starting to develop with little to no planning"* (5).

Customer expectations significantly affect project success as this influences customer satisfaction. Hence, any expectations that cannot be met need to be communicated. The interviewees mentioned that some expectations could not be met due to a lack of resources at the organisation or the customer's expectations not aligning with the Agile way of doing things. Interviewee 5 explains, *"From a project perspective, I think being realistic when you plan is imperative. It is important to compare those expectations to resource availability and determine if they are realistic. Otherwise, the project is doomed to fail from the get-go because the expectations are unrealistic".*

Customers need to understand what Agile software development entails, as this affects their perception of the process and can be the basis of their expectations. The

organisation must ensure a customer understands Agile. *"Everybody is used to waterfall, so when you deliver something, they expect you to deliver the final product"* (4).

Commented [DH8]: Doesn't make sense - rephrase?

The organisation is responsible for explaining the process to the client to manage customer expectations. "*We need to make sure that the customers and the business sponsors, before we even start the project, so, during the feasibility stage, they need to understand already that we will be doing Agile and what that means and the implications of staying agile*" (4).

A customer's lack of understanding can lead to them wanting to see progress in cycles that are too short for the team to deliver substantial work, or they can expect the first deliverable to be the final product. Interviewee 13 details customer expectations' impact on decision-making and the quality of work: "*Okay, so agile is obviously to see…rapid change, right? However, it is not built for a client to see progress every week...The issue I have seen with dev teams is that the client will want to see something prematurely before everything is thought out 100% and set up correctly. So, developers will make rash decisions; even designers will make rash decisions…then you end up with inconsistencies as you continue." (13).* Enough time must be given between delivery cycles so the project team has enough time to review the deliverables and produce quality work.

On the other hand, some clients may be under the impression that the first deliverable is the final product and end up disappointed or taking that as the final product and not making any improvements. *Depending on the client's understanding of Agile. It can be an issue if you can put something out pretty quickly, MVP or show them progress. And they are like, okay, cool. So, this is it, right?" (15).*

Hence, to manage customer expectations, it is important to communicate to the customer, the sponsor, the user or the customer representative what they should expect and why from the start. *"Start setting like the ground for realistic expectations. Yeah, and what that means is saying, hey, if you want this, we can do it, but it would probably take a lot more time and cost you more and here are the alternatives"* (6).

5.2.2.2 Communication

Communication was ranked seventh in the frequency-based ranking, and it is the third highest-ranked organisational factor. One interviewee stated, "*Communication is one of the biggest factors within teams and the organisation" (14).* A common thread in the findings is that the interviewees emphasized that communication must be effective. For communication to be effective, this entails having the means to communicate in real-time or using methods that allow for quick responses. Any delays in the responses can mean delays in the project. *" Communication is really important…having methods and techniques in place to facilitate clear and as close to real-time communication as possible… what really determines success or failure often in terms of communication is how comfortable it is for people to communicate in real-time." (5).*

The case study organisation has some individuals working remotely, and the communication methods must cater to this. *"With Agile, communication is open and easy. Nowadays, with remote working, the organisation needs to ensure that there is a way for people to communicate effectively"* (4). What Interviewee 4 mentions may apply to all Agile organisations with colocated teams. A method to aid in effective communication, particularly online communication, that was brought up is using a specific tool throughout the organisation for communication. Interviewee 8 states, *"I will, from my side, use ClickUp and Slack. It has been a massive help in organisation and communication, so I see the need for them"* (8). The primary communication tool used in this organisation is Slack. However, organisations can pick the tool that suits them best.

Some important things to communicate are any challenges individuals face with their work and updates on their work. *"So, communicating between team members on what is working and what is not working is good*" (15). Interviewee 3 explains that team members need to be willing to ask for help when needed. *"I think it is also the communication between team members. Willingness to say, like, listen, I am, I need help, would you be able to sit down with me for 20 minutes?"* (3). Both statements echo the idea that team members must keep their team members informed of their work and how it is going.

The communication factor focuses on internal communication between organisational and team members since client communication is discussed under the customer involvement factor. However, it is significant to note that an organisation is still responsible for facilitating good external communication from the organisation or team to customers or partners. *"So, if I have to speak to first my manager, he has to speak to the manager, and then the CEO, only we will get to the client. So, that is not a good process to follow. We should communicate openly with the client" (9).*

External communication includes informing the customer of project progress and any organisational changes that affect them. *"Having a good sense of communication is important. Where is the project going? How is it going? Are there any roadblocks, updates or extensions that need to take place? I think maybe that is also one of the things we, well as an organisation, should try to communicate with the client" (6).* Both the clients and the organisation play a role in establishing good communication.

5.2.2.3 Organisational Culture

The organisational culture may be characterized as a collection of organisational elements or variables impacting a company's agile software development (Aldahmash et al., 2017). According to the findings, organisational culture is the most influential critical success factor as it was the highest-ranked. *"I would say that the organisational culture is very important. How the culture is within the organisation, and if it benefits everyone"* (9).

Interviewee 5 gave an example of a culture that would not work in agile software development, which is a competitive culture: *"You need to be in a place where it is about the solution at the end of the day and getting the best solution and not about scoring points as an individual. I think it comes down firstly to culture… I think culture is a critical thing. If you are working in an agile environment where everyone is trying to prove that they are better than everyone else, it will be tough to do the job" (5).* Such a culture would not be advantageous to a project.

An advantageous organisational culture would have the following attributes mentioned by the Agile practitioners: adaptability to change, knowledge sharing, a suitable organisational structure, and organisational buy-in into using Agile.

In agile software development, it is critical to have agility. An organisation with a culture that is adaptive to change can quickly acclimate to new ways of doing things, as explained by Interviewee 2: *"If an organisation has a culture of adapting to change, that is very important for agile because, especially with agile, everything changes quickly. If an organisation is resistant to change and stuck in the old ways, it is difficult for them to change to a better system or upgrade for anything."*

Another aspect of organisational culture is knowledge sharing. It is perceived that *"Organisations that focus more on the learning and advancement of the employees will lead to their employees producing better work"* (9), and it can be inferred that this may result in better quality deliverables in a project. A knowledge-sharing culture needs to be cultivated in an organisation, and as mentioned by Interviewee 2, *"This might relate again to organisational culture, having a knowledge-sharing culture. So, for instance, we have a knowledge share opportunity at an organisation we work at ... It is amazing because just cultivating a culture where people are hungry to learn and creating the environment for them to do a knowledge share, more so in virtual teams*" (2). Therefore, there needs to be opportunities for knowledge-sharing created in an organisation.

The following attribute of organisational culture is the organisational structure. According to the findings, a flat organisational culture with less power distance suits Agile software development. *"I think organisational culture is a critical thing … In terms of organisational structure, I think it needs to be flat in the sense that there is not a strict chain of command. In an agile environment, the best ideas need to win … it is important that everyone's ideas matter and that no hierarchy goes against that."* (5)

An organisational structure with fewer hierarchies reduces bureaucracy, which means delays in decision-making in the project are quick, and work can be done more rapidly. *"So, depending on how large the structure of the specific project is, we have a structure*

in our organisation, but that usually informs the structure of the project... So, if there is like a lot of bureaucracy or if there is, like, a massive reporting hierarchy for one project, it could be quite difficult to work with" (1).

Lastly, an organisation needs to buy into the idea of using agile, which goes hand in hand with ensuring the organisation understands what agile is. It will be challenging to successfully use an agile methodology in a project without knowing what it is and its implications on how a project is carried out. *"Sometimes organisations believe that agile means no rules, and we can make the rules as we go. So, it is a good thing for organisations, management in organisations, to understand that, yes, we need to be agile, and we need to be open to accepting changes, but that does not mean that we will allow scope creep. It does not mean that we do not have a deadline."* (4)

5.2.2.4 Facilitative leadership

According to agile practitioners, the leaders in an Agile software development project must be facilitative. *"In an agile environment, you do need strong leadership, you know, to guide the team, as the nature of leadership should be one of facilitation and enablement rather than autocracy"* (5). Facilitative leadership in an organisation is described as management that is open to suggestions and facilitates and does not dictate how a project is carried out.

A facilitative leader trusts the team to work without much intervention and close monitoring. Interviewee 3 explains this: "*Trust that your team has the skills to do what they need, management can jump in if necessary, but other than that, sort of leaving you just to do your job and like not police you the whole time" (3)*.

A facilitative leader empowers the team to make decisions without running everything by management or team leads. "*You should trust people to do their jobs. So, developers should know what they are doing. You must trust them to do what they are paid for. It should not be that everything should be discussed, and everything should be designed. People have a core skill, and you should trust them to do their jobs" (10).*

However, being a facilitative leader does not take away the responsibility the leader has to manage the team. They still need to take accountability with the team for any failures or successes. "*When managing, you should be able to take accountability and responsibility without putting all that pressure on the people just doing the execution. Because in most cases, if a project fails, the management might be like, oh, well, but these guys are not even working. However, if you manage correctly, you can pick up those things way earlier on the project*" (9).

A facilitative leader is open to ideas and suggestions from the team on improving anything related to the project. "*There should be no dictatorship. If I need to speak out about something, I should be able to. If I have ideas, I should be able to present ideas*" (9). The management or team leads must create a safe space for team members to speak up and share ideas. *"Yeah, so creating a culture of sharing ideas, a safe space for people to communicate if something bothers them"* (6). When the ideas are shared, they should be considered and not just heard but not given a second thought. Interviewee 12 advises, *"As a manager, you should consider other ideas before putting yours as the main proposal. Do not be headstrong on your ideas, but be more open to different solutions to accommodate an issue."*

Commented [DH9]: Missing word/s

The leadership style used in a team plays a substantial role in how the project turns out. "*So, if the people managing the project are good, it works fantastic. If they are not, things can quickly unravel, affecting everything else in the project."* (11)

The literature in the SLR identified a manager with an Agile, suitable management style as a critical success factor that may be similar to this. However, according to the findings, facilitative leadership is not just about management but also the other leaders that can be found in an agile project, such as a team lead, technical lead, design lead or developer lead. Thus, it is poised as a new factor. The leaders in the organisation who have a facilitative approach and exhibit the attributes discussed in this section may improve the chances of project success.

5.2.2.5 Conducive and collaborative work environment

A working environment relates to working conditions and operating processes that include communication, coordination, and collaboration among members of a distributed agile team (Tyagi et al., 2022). In the organisation, some team members work from the office, others work from home, and others alternate.

Individuals work differently, and the organisation prefers people to work in an environment conducive to productivity. *"People are different. It is a more conducive environment for productivity for some people working from home, like me. However, working at the office for others will be more conducive. They would be more productive in that environment"* (1).

According to the findings, a healthy environment allows for online and face-to-face collaboration. *"In Agile, working together and collaborating is really important"* (4). If team members are in different locations, it came up that there are still some challenges with communication and collaboration. *"Also, personal opinion, I think that this whole working from home is not working. The communication is key. Especially because I have struggled to communicate with people"* (8).

If individuals work from a different location to another team member, they need to use the appropriate tools and technologies to enable collaboration. *"There are people without the infrastructure to stay online during load shedding. So, you know, if a person is unavailable for two to four hours a day, during working hours, and there needs to be communication, meetings, or deliverables done, that could affect a project greatly" (1).* Unavailability can cause delays in a project and build frustration between team members.

Interview 1 states, *"It is ideal for people who thrive in a particular environment to be in that environment that they thrive in".* However, if working in separate locations does not benefit the team and hinders collaboration, it would be more beneficial for the team to be collocated. *"In a lot of the cases, if a company does not have*

measurements in place that ensure effective online communication, then the success of the project depends on the proximity of the individuals working on the project" (2). Some project-level measures can also be taken to enable collaboration within teams in the organisation, such as having daily stand-ups and sprint retrospectives. Daily stand-ups allow the team members to update each other daily on their work. *"I have seen in the organisational factors of things, having a stand up every morning helps communication. It helps to understand what roles people are fulfilling" (8).*

Sprint retrospectives allow the team to give each other feedback on work in progress or completed, as stated by Interviewee 15: *"So yeah, so I think retros can be super useful for ensuring that you know where you are at, what has not been done, why and then as a place to give feedback on what we could be doing better"*.

5.2.2.6 Societal culture

Societal culture is a system of shared beliefs, values, and ideals learnt, passed down through generations, and represented in society's laws, policies, and behaviours (Tam et al., 2020). Suppose a project is being developed for a target audience with a different culture than the organisation developing it. In that case, societal issues may arise that affect the project. *"I think I feel like the diversity of the group working on the project is important because, especially in South Africa, there are different LSMs, different cultures, and different backgrounds" (*2).

One of the possible issues is language barriers between team members and clients. *"And then language barriers. I have had a couple of experiences. What they expect and what we understand is not the same thing. English is not their native language. So, it is not their business language. That influenced a couple of projects"* (11). Language barriers can hinder communication, leading to misunderstandings that may affect project success.

This factor was the lowest-ranked critical success factor in this study since only two agile practitioners mentioned it in the interview. Although it was identified in the SLR as a critical success factor, the study (Chiyangwa, 2017) did not identify societal

culture as a CSF of Agile software development projects in South Africa. Thus, it can be argued that societal culture may have a minimal influence on Agile software development projects in South Africa.

5.2.3 Team factors

The team category covers factors relating to the people who manage and execute the project (Arcos-Medina and Mauricio, 2020, Chow and Cao, 2008, Hummel and Epp, 2015, Misra et al., 2006, Misra et al., 2009, Shakya and Shakya, 2020). This includes the team members' characteristics and properties necessary to complete the task (Hummel and Epp, 2015). The team category also examines practices relating to the conduct and working style of those participating in the software development process (Arcos-Medina and Mauricio, 2020). The combined descriptions above were used to determine the factors to place in the team category in the framework. These factors are team capability, coherent – self-organising team, small team and team morale. These factors are discussed below.

5.2.3.1 Team Capabilities

Team capability refers to using knowledge and the conditions that enable teams to do their jobs successfully (Tam et al., 2020). This factor was the second highest-ranked team factor in eighth place in the overall ranking. According to Chow and Cao (2008), the attributes of team capability include competence, expertise, motivation, technical training, managers with agile knowledge, and adaptive management style. The attributes of team capabilities identified in this study are very similar: technical skills, expertise, technical training and project managers' experience.

According to the agile practitioners interviewed, a team's skill set significantly impacts whether they can deliver a project and do so on time. *"So, there are skill constraints. We often talk about needing to deliver this work in this amount of time, which is imposed on the team. Often, the team's skill set and the tools at their disposal are not always considered. So, I think skill set is an important factor*" (5). Thus, as highlighted

by Interviewee 2, the skill set of a team needs to be considered before the organisation takes on a project.

In addition to technical skills, the team members need expertise in the project domain and tools and technologies to be used in the project. *"First of all, proficiency with tools or platforms that we use. The more proficient you are with it, the faster you can do something. That will help turnaround times on feedback"* (11). As noted by Interviewee 11, if the individuals in a team are proficient in the tools and technologies used in a project, this increases the rate at which work is done and at which work can be presented for reviews to get feedback.

Hence, the team's skills and expertise must be relevant to the project; otherwise, they might not be as useful. *"Choose the team based on the requirements; you should make sure that you are a team that can suit all the different requirements that will be in the system but not limited to one position, so if you have issues, more than one person is able to assist" (12)*

The team might need technical training before starting the project if there is a shortage of the required technical skills. *"You might have a technical skills shortage, which we also need to plan for training, looking at your technical team, and deciding if they need training and what kind of training they need?"* (4). However, if a lack of technical skills is discovered during project execution, the team might need training as they work on the project. "*So then maybe let us say a task is very complex, and the people are executing, and they struggle. Management should be open to amending how they manage the project to get more time or help, such as more training*. (9)

Furthermore, not all individuals can have the same expertise and technical skills; thus, the team's combined skills must be considered. *"…to know everyone in the team's limitations, strengths, and weaknesses to help each other improve. So do not expect more back-end developers to do the front-end, or people with limited knowledge or lower-level experience to try and develop intricate software as that*

might take them longer" (12). The team needs to be composed so that their skills complement each other and each person adds value to the team.

One way of having a good team composition, as suggested by Interviewee 2, is having a good distribution of levels of experience and expertise. *"Having that effective spread… juniors, intermediates, and seniors. So that your knowledge spread is equal and grows as the project continues, there are enough people to do reviews, and there are enough people to do the menial tasks, like if a feature breaks to fix the bug." (2)*

Lastly, the project manager's expertise also affects the team's capability. If the project manager assigns work to individuals based on their skills, expertise and strengths, this increases the chances of success. Interviewee 1 explained how having an experienced project manager's experience in a team positively impacted a project. *"The project manager's experience is surprisingly vital. In my first year of work, we did not have a set project manager. And then, a year ago, we got a project manager and the difference in how things were managed and assigned. It just feels a lot more balanced and a lot more sane."* (1).

5.2.3.2 Coherent - Self-Organising team

The interviewees mentioned that the project team must be self-organising and not reliant on management. In the agile approach, a self-organizing team can coordinate their work independently and entirely control the development process (Stankovic et al., 2013). The attributes of a coherent self-organising team identified in the finding were team cohesion and coordination.

Coordination in a self-organising team involved deciding who would work on what tasks without relying on management to tell them that. Interviewee 6: *"I think coordination is a big one as well. Do we coordinate ourselves in such a way that we are not working over each other?*" (6) The team needs to be able to organise themselves such that they are not working on duplicate tasks, or some tasks are left unassigned. If a team cannot coordinate, *"you get duplication of efforts because people are not telling each other what they busy with,"* as interviewee 4 stated.

For a team to be effectively self-organising, there needs to be cohesion. According to the interviewees, how well a team can work together greatly impacts project success as it can affect other aspects such as team morale and productivity. *"The biggest thing that can have an impact on project success is how people work together."* (10)

Team cohesion has much to do with the people's personalities in the project. Some personalities can work well together; it is more difficult for others. *"The cohesion of the team as well. So, I mean, we try not to let interpersonal issues get in the way of projects. But you know, some personalities just do not work together. Yeah"* (1).

In a cohesive team, there is respect between team members. Interviewee 9 states that there needs to be respect among team members. *"The team should have some sort of respect towards each other"* (9). Additionally, interviewee 12 added to this, mentioning that the team needs to know each other strengths and weaknesses and how they can be used in the team. "*Respect for one another, to know everyone in the team's limitations, strengths, and weaknesses to help each other improve"* (12).

Lack of cohesion causes delays in the project, and if a team can work cohesively, this can increase productivity. Interviewee 13 cited a situation where he could not work well with a team member and how it negatively impacted the project. *"So I have seen split, a real stepping on each other's toes, like, and I had this issue in one of my earlier projects, I mean, that the developer had very conflicting ideas about how things should work and ended up hindering the project…The project would have come out faster than if you had put us together and tried to get us to make the best solution because we could not work cohesively"* (13).

5.2.3.3 Small Team

This factor relates to how a small team is more suitable for an agile project. The team size commonly cited by the interviewees as a small team was a team with four to five members. *"A good average team size is around four to five,"* says interviewee 7. It

was also noted in the answers to the background questions, as shown in section 4.3, that 40% of the interviewees were in a team with five to six members, which supports this notion.

According to the findings, a small team is easier to coordinate and manage. Interviewee 10 points out, "*Yes, I think too big or too small teams do not work. But that is dependent. So, if a team is too big, then it is difficult to organise the team"* (10). Interviewee 5 expressed a similar perspective; *"From my experience, it is ideal to have no larger than three to five people, depending on the project, but anything bigger than that starts getting difficult to manage"*.
"Having teams too large complicates figuring out who is responsible for what. It blurs communication lines. I think one primary important aspect of Agile is that information needs to flow effectively. If too many people are involved, you often get confusion, miscommunications, and misconceptions" (5).

Additionally, agile practitioners believe that working in a small team increases the chances of an individual having a sense of ownership of their work. Interviewee 13 explained, "*The smaller your team is, the more, the more they have ownership of what they are making" (13).* The interview elaborated on how being in a larger team made them lose the sense of ownership for their work: "*I started getting less enthusiastic about work. I started not really caring that much about the product that I was making because I did not own any piece of that product" (13).* As discussed in the next section, a sense of ownership was identified to affect an individual's motivation and influence team morale.

Thus, the benefits of a small team cited by agile practitioners that make it suitable for Agile software development are easier coordination and communication, more explicit responsibilities, and a greater sense of ownership.

5.2.3.4 Team morale

According to the frequency-based ranking, team morale is the factor with the most influence. Overall, it was ranked in fourth place. The SLR has a critical success factor - "Team motivation"- similar to Team morale. However, the interviews mentioned

motivation on an individual level as an attribute that influences team morale. Thus, the researcher decided to stay true to the interviewee's terms and refer to this factor as team morale. *"The sort of mindset and attitude and culture and morale of the team can have a big impact on a project … It is important that morale is kept intact, and people believe they can still deliver on the work is an important factor"* (5).

Interviewee 1 explained that team morale involves the team believing in the project and wanting to complete it, *"So, things like team morale, the team, just like them believing in the project or them wanting to complete the project. I think that falls into morale as well"* (1).

These factors mentioned along with morale were attitude, mindset, and drive. These aspects of team morale are perceived to affect project success. So, for instance, the team having a positive attitude, a similar mindset, and drive would affect the project positively, as mentioned by interviewee 3. The interviewee states that *"Having a team with a can-do or a positive attitude does wonders and teammates like willing to help out…everyone has the same mindset, the same drive to get things done, the same push, you know team attitude would like fit under this" (3)*

Team morale can be affected by several things that project management needs to be aware of and pay attention to in a project. Changes in a project can affect the morale of a team. Interviewee 2 gave an example of a situation in which changes affected the team morale "and also *the morale of the team the one week, we spent an hour trying to figure out how to follow all the steps in a process, and then they changed it again, it is it gets very discombobulating for the employees."* (2) Thus, changes in a project need to be managed well as the interviewee explains further, *"If a person is unhappy with an applied change, if you do not manage that, they will leave, and then you are stuck with fewer resources which affect your triangle."* By triangle, she explained she meant meeting scope, time and budget.

Another aspect that affects team morale is the motivation of individuals in the team. *"In terms of motivation, if a person in a team of a person slacks, then that usually creates bigger issues for the team."* (14). The interviewees mentioned some ways

individuals can be motivated in the interviews. The first is through the compensation or rewards the organisation offers, *"For some people, it is also just a matter of, like, how motivated they are to get compensation"* (1). Secondly, Interviewee 1 also mentions that a reward system is a factor that motivates team members: *"Usually when a project goes really well, you do get a reward in terms of words of affirmation, which usually translate into more respect, more opportunity for going up in the hierarchy".*

Thirdly, others are self-motivated, *"A characteristic I think the people executing should be able to have is they should be self-motivated*" (9). Lastly, for others, motivation stems from a sense of ownership over one's work: "*They should also take pride in their work and have ownership. That is a characteristic I think the people executing should be able to have. They should be self-motivated as well"* (9).

Team management also needs to monitor the well-being of the individuals in a team and manage it accordingly to maintain team morale. *"Management of the team must be aware of the mental health of the people working because if one person breaks, it can affect the morale of everyone, and this person will burn out…"* (2).

If some team members experience "burnout", this can affect the whole team's morale. Interviewee 3 mentioned a way to avoid burnout for the project management to ensure a balanced work distribution*: "How the project is managed matters… before some people would get a lot more tasks, and some people would not get enough. So, you know, that would usually cause burnout and affect some other aspects of the team. Now that a project manager manages this, things are usually evenly distributed" (*1).

According to the findings, team morale is a CSF that needs to be monitored and managed closely throughout a project. The team is influenced by several aspects, as highlighted in this discussion, with the most outstanding being the team members' motivation, mindset, attitude, and drive. An organisation needs to develop projects with motivated people, providing them with the conditions and support they require and trusting them to do their work well (Fowler and Highsmith, 2001). More aspects

not identified in this research might affect team morale. This research did not aim to find the aspects that affect team morale but identified some in the interviews.

5.2.4 Customer factors

Customer factors include factors relating to the people who sponsor the project or will use the product (Arcos-Medina, 2020 #183; Hummel, 2015 #125; Shakya & Shakya, 2020). The customer factor identified in this category was customer involvement.

5.2.4.1 Customer Involvement

The agile practitioners interviewed in this study expressed how important it is for a customer or customer representative to be involved in project activities, as this drastically impacts the project's success. "*When a customer or the project sponsor is engaged in the project, the chances of success of their project are just a lot higher*" (10).

This factor was the only critical success factor identified in the customer category, and in the rankings, it tied for second. According to the findings, the attributes of customer involvement are user involvement, customer feedback, direct communication, and customer buy-in. These attributes differ from those initially identified by (Chow and Cao, 2008): strong customer commitment and presence, good customer relationships, and the customer having full authority.

The interviewees observed that when a customer has a hands-on approach to a project, that project is more likely to succeed because the customer will be available to provide the team with the information, feedback and help needed to meet the requirements. *"The clients where we have really made success is with clients that are hands-on, you know, they are like, hey, listen, where can I help? Where can we assist?... What access do you need? We may have to sit down and meet about this ...feel free to contact me at any time. I will get back to you as soon as possible - that type of thing."* (3)

The customer may be the project sponsor, a user, a representative of the organisation or all three. Each role would need to be involved in the project to increase the likelihood of project success. *'The business sponsor also needs always to be reachable and*

available to give the input because that is another thing that can break a project is when clients are not involved in the actual development process" (4).

The business sponsors' or their representatives' role includes confirming if the project deliverables meet the expected requirements and signing off on them. If a representative is sent, they must have context about the project and know the requirements to be met. "*If the clients somehow send a proxy to sign off on their side, so if we say okay, we are done, we hand it over to the client, and they sent a proxy, or another person not involved in the project's inception. Then, they might miss the mark and sign off on something wrong"* (1).

The user's role involves providing user requirements and giving usability feedback. Sponsor requirements are not always the same as the user requirements, so it is sometimes necessary to get them. *"Hence, a client, one of the issues that I had in the beginning, was a client may have a certain set of requirements, but a user could have a completely different set of requirements"* (1).

Agile practitioners emphasise user involvement because the software user can provide unique feedback on deliverables; accordingly, they must be involved where possible. "*If you never actually look back at what you have done before and ask the user, is this iteration good? Are we making improvements? Then you are just making many changes. You are not actually like improving on anything. So that is where involving the user can be good"* (15). As detailed by the interviewee, user feedback can be used as a benchmark to determine if the customer is satisfied with the parts of the system that have been delivered, contributing to project success

The interviewees stressed the significance of customer feedback on their work as this is what they use to make improvements: *"Availability, like are you available to do tests and give feedback when we deliver a minimum viable product?" (2)*. However, this is not all the project team requires from a customer. In addition to giving feedback, the customer needs to be available to respond to messages or emails, attend meetings, test the system, provide resources or data needed, or put the team into contact with the

relevant people. Thus, client availability is essential for project success. *"I have seen so many projects fail just because of the unavailability of customers*". (5)

Customer involvement also includes putting the team in touch with the person who has the information the project team requires. Suppose the project team requires information from another person in the customer's organisation. In that case, the team should interact with that person directly. *"The project where it is not a success is when the client tries to play a middleman. Direct communication with developers makes a big difference, rather than going through this middleman that is not technically inclined"* (3). The interview calls it direct communication, a term often used in literature. Stelzmann et al. (2010) say having as much direct communication as possible between customers and development is vital because direct communication is efficient and incurs fewer losses.

Furthermore, interviewees stated that a customer needs to value the project being undertaken. They emphasised that when a project is valuable to a customer, you can see it through their involvement. Interviewee 10 explains that *"many customers just throw money at the problem, and then they never engage themselves. But then the message that sends out is that that project is unimportant."*

The value the customer places on the object has been observed to affect the time they take to respond to the project team. Interviewee 3 explains this using an example from a current project: *"In the project that is not going so well right now, it almost seems they do not want the product they hired us for; there is zero drive from their side. When we try and communicate with them, it takes them a day or two days to get back to us, and it really slows down progress"* (3). So, customers must promptly provide feedback and responses to reduce project delays and keep the team's momentum going.

The customer involvement factor is a rich factor that encompasses quite a lot of aspects that are critical to project success. This factor was the only identified factor in the customer category and was ranked as the 3rd most important critical success factor. This emphasises the importance of customer involvement in a project. Customer involvement is vital in agile software development (Aldahmash et al., 2017).

5.2.5 Process Factors

The process factors category covers factors relating to how project activities are carried out (Arcos-Medina and Mauricio, 2020, Chow and Cao, 2008, Hummel and Epp, 2015, Misra et al., 2006, Misra et al., 2009, Shakya and Shakya, 2020). The following process factors were identified: flexible process, accurate progress tracking, accurate planning and estimation, process clarity, continuous delivery and proper work handover.

5.2.5.1 Flexible process

Flexibility is the essence of agile software development; thus, it is unsurprising that the agile practitioners in this study mentioned a flexible process as a critical success factor. This factor is not identified in the SLR as a critical success factor, and this may be because it is assumed that the process should be flexible in agile software development. However, this study explicitly states it because of the importance placed on it by the interviewees. This factor aligned with one of the values of Agile detailed in the Agile manifesto: "*We value responding to change over following a plan*" (Fowler and Highsmith, 2001:2).

This factor was ranked seventh; according to the ranking, it is the second most crucial process factor. The attributes identified for a flexible process were aligning the process to requirements and getting team feedback about the process.

According to the findings, the process used in Agile software development needs to be adapted based on what is required in a particular project. *"So, each project is also unique. You cannot copy and paste the process that worked for a different one onto another and expect it to fit. It needs to be adaptable to each specific project because each project will have its nuances"* (11). Organisations need to be able to adjust a process to the specific project as this increases the chances of success. *"The process needs to be correct and adaptable to projects, and the differences between different projects…because obviously, not every project is exactly the same"* (11)

Flexibility does not mean starting a project without a process and piecing together one as the project goes. The process can be identified at the beginning of the project, but if it does not work, it is better to change aspects of the process to fit the project. *"I think it is essential to establish your process, and yes, we can change the process during agile, but we need to establish at least a baseline process in the project's beginning or foundation phase."* (4)

The adjustments can be made by identifying which activities will be carried out in a particular project from the usual, depending on the budget, schedule, and scope. Some activities are core and a part of every project, but even these can be scaled down or up depending on the project. In addition, other activities may not be necessary for all projects and can be removed entirely. *"In my opinion, from our consulting experience, not being dogmatic about the process. This may be troublesome in the academic world, but being pragmatic and adapting the process depending on the scenario to keep the core pieces and add pieces, if you can, would be beneficial. However, priorities are often very different for particular problems or scenarios where there may be a tighter deadline, or it is a lower-risk project. So, you cannot have a one-size-fits-all process." (5)*

One way to know that the process is not working is when the team fails to meet deadlines or does not deliver the estimated amount of work. *"The process should be a facilitator and enabler of a coherent, systematic work, as opposed to adding unnecessary administration on top of already stretched people" (5).*

The project managers or team leads can get feedback from the rest of the team on what needs to be improved when a process is not working. "*It is a vital part of the process to do it in a way where people actually give feedback. I think this is something that we can currently improve, getting feedback about the process*" (15). Even if management or team leads think a process is working, getting feedback and being proactive about implementing changes is essential.

Once the feedback is received and the changes implemented, they must be communicated to the team. The change may be gradual as the team might need time to adjust; thus, the sooner the failing process is adjusted, the sooner the project can get

back on track to success. *"I think being a software house, we should be open. If a process does not work well, be open to changing that process and allowing enough time for staff to adjust to this change process."* (2)

5.2.5.2 Rapid Feedback

According to the interviewees, feedback improves the quality of work produced and helps the team know whether they are on track to meet the requirements. Interviewee 6 observed: *"It looks like you develop things better when you have better feedback cycles. So, in my mind, often when I think about this, it always feels like just a conversation about optimising our feedback loop. So that you know at the earliest point whether you need to change something or whether it is successful and meets the goal"* (6). Thus, feedback must be given often and as quickly as possible.

The interviewees cited three points where feedback is usually required: top management feedback, reviewer feedback and customer feedback. However, customer feedback is discussed under the critical success factor, Customer Involvement.

A project team sometimes delivers outcomes that need top management sign-off before they continue. "*An organisational factor contributing to a successful project is overall admin turnaround time. So, if things need to be sorted out by a project manager or administrator, like where things get handed off to other teams or organisation members, and you know, you need those things to return to be signed-off before things can keep moving"* (1). In this instance, delays in deliverable sign-off mean the team members are blocked and cannot continue with work, which may delay the project.

Team members deliver outcomes to reviewers or team leads or for feedback on work in progress to make improvements. *"I think having a chain of reviewing is also a good review cycle and the availability of reviewers. The higher up you get in the food chain, the busier you get, but these people now have to check up on the people at the bottom. So that also creates a delay in the process"* (2). In this second instance, the sooner

feedback is given, the sooner improvements to current work can be made, resulting in improved deliverable quality.

An interviewee mentioned that with team members, feedback does not always have to be provided in formal ways. *"Another thing is to get the technical feedback as soon as possible; do not wait for the next meeting. Have chats, ask quick questions, and put comments in the work. Things like that move much faster than waiting for next week because that is when we have our next meeting with the developers."* (15). Thus, the project team must use whatever means to facilitate quick communication to get feedback.

5.2.5.3 Accurate progress tracking

Progress tracking is helpful in Agile software development because it gives a good sense of how far the project is at all times. Progress tracking, if done accurately, is beneficial for giving customer feedback and assessing if the progress will be completed on time. If not done accurately, it does the opposite, as explained by Interviewee 4: *"So people can have the same task, but none update the board to say that I am working on this task or none of the new tasks gets on the board, leading to errors in reporting and misconceptions about where the project is".* It can be extrapolated from this observation that the team members must update the ticketing system for the progress tracking to be accurate.

The organisation in the case study uses what they refer to as a ticketing system, and the one they use is called "ClickUp. *"The ticketing system makes a big difference, and finding the right one that works is Yeah, it is good currently"* (15). They observed that this system is suitable for them and assists them in accurately tracking the progress of their projects. *"The ticketing system, I think, works quite well when implemented correctly. So, at the start, it can be daunting seeing all these tickets on the board… It becomes easier as you start marking off one after the other. You can see actual progress.*" (3). However, an organisation can use a ticket system that is suitable for their organisation.

5.2.5.4 Accurate planning and estimation

According to the frequency-based rankings, accurate planning and estimation is one of the most influential factors ranked second and the most influential process factor.

Jintian et al. (2022) state that many software development projects fail because of a lack of strategic and tactical planning. So, like in any other software development project, planning is also required in Agile projects. *"How the management handles the planning is very important"* (9).

According to the Agile practitioners interviewed, and as stated in Chapter 2, planning in Agile is not as detailed as in traditional software development. *"So agile is often not associated with planning, but there is planning on some level. So planning is, to me, still relatively important. You do not have these massive, long-winded months and months of planning. You do not have to plan in the finest level of detail (5).* As stated in the Agile manifesto in the agile methodology, we plan, but we acknowledge the limitations of planning in a volatile environment (Fowler and Highsmith, 2001)

The planning may not be detailed, but it should be sufficient to provide all the parties involved with a clear goal and responsibilities. *"You need to plan such that all the stakeholders and parties involved are more or less on the same page, and the people that need to do work are clear on what they must do" (5).* Interviewee 9 echoes the same notion: "*So, if the management has effective planning, then we know the clear goals, we know what we were going for, what needs to be done"* (9). It is important that in the planning phase, the responsibilities of the team members are laid out as expressed by Interviewees 5 and 8 ."*I think it is very important, and then just understanding responsibilities, like where your responsibility, your primary responsibility starts and stops" (5), "It helps to understand what roles people are fulfilling" (8).*

The interviews also indicated that the task must be turned into a ticket before the task can be done, which is part of the planning for a project. Tickets represent work units; they should contain descriptions of the work the team is expected to complete

(Thomas, 2008). The accuracy in detailing work that needs to be done in a ticket and the information on the tickets will impact how the work is done and implemented. "*If the project manager or whoever is creating tickets describing the work that needs to be done is ill-informed on how developers consume this information, that affects the success of the implementation of the ticket"* (2).

If the tickets are not comprehensive, this also hampers progress as the person working on the ticket would need to interpret the meaning or try to find more information. *"I also think that tickets need proper descriptions … it does help a lot because if it doesn't, you sort of need to try and like infer what this person wants, like if it is a one-liner, "implement this thing", you need to infer what that means or whatever. That can be very daunting and hampers your progress"* (3).

Interviewee 2 explains that a comprehensive ticket includes the following information: "*What is the context in which this feature is used? What is the existing subsystem that I am working on? Is there anything that has already been developed that affects this or that I can reuse to do this? Everything related to this work should be included. So, it should be a comprehensive ticket, including related tickets and where to find external information I can use on this ticket*."

Interviewee (9) called it a "well-defined task", and when asked what a well-defined task includes, they gave the following: "*What the requirement is targeting, then what is the goal of that specific task? … And then, what is the expected outcome of the task? I think those are the things that could be contributing to a well-defined task*."

Lastly, planning entails accurate time estimation of the tasks to create the project schedule. *"I think the pre-planning is critical. I believe it is crucial to ensure you are allocating the correct amount of time to tasks" (8).* An inaccurate time estimation can cause the project to be done in a rush and produce poor quality; as Interviewee 14 explains*: "Properly calculating how long it will take to complete a task becomes important because that properly marks the quality you will bring and are how fast you can get it out. The quality might be downgraded if you calculate it too low."*

One way to estimate time accurately in a project is by using previous projects as benchmarks. *"So yeah, in terms of timeline, I suggest that the only way you can predict reliably and efficiently is by getting a gauge of project size and then using your previous project timelines as a record, especially if you build projects the same way" (13).*

Another way to make time estimations more accurate is to ask the team members, as explained by Interviewee 15: *"So I think when planning with estimations that's where the team could be helpful is by stepping in and saying, hey, I think that this could take this long or helping you if you are not sure about how long something should be estimated for."*

Thus, project management needs to make an effort and use whatever means at their disposal to ensure that the planning and related activities are done as accurately as possible.

5.2.5.5 Process Clarity

When a process to be followed in a project has been decided on and set in place, it is key to communicate it to all the team members. *"The other factor is process clarity, so making sure everyone is on the same page regarding the process" (4).*

If the process is clear to all team members, it is easy to follow, and the whole team can stay aligned. Otherwise, the misalignment can cause chaos, as detailed by Interviewee 15: "*If everyone is clear on what they are doing and the process, it makes a huge difference. I have seen teams like that where half the team is following it, and half the team is not. Yeah, and it was just chaos."*

Interviewee 11 says, *"If done well, proper process is in place, and everybody understands the process. It helps make the project faster and helps smooth things along, and if there are problems, they can be quickly resolved within the process."* Consequently, if the process is clear, it may help a team complete a project quickly.

5.2.5.6 Continuous delivery

Continuous delivery is an approach to software development that delivers new features to customers as soon as they have been built and tested.(Lehtonen et al., 2015) According to the findings, a continuous delivery approach is believed to make an Agile project more successful. *"If you try just to deliver some things in small pieces incrementally and test those small pieces, that increases how successful your Agile projects will be"* (6).

It is also beneficial for the project team to provide opportunities for a client to give feedback on the software being developed as frequently as possible. *"A project can deviate from what the client intended in several ways. So, if you minimise the time between getting feedback from the client and showcase where we are and what we are doing regularly, that often helps"* (6). The interviewee has observed that frequent short feedback cycles help the project team to stay aligned with the requirements and the customer's expectations.

The Agile practitioners mentioned that they usually deliver a minimum viable product (MVP) to a customer as the first software deliverable. An MVP is a version of a product delivered to clients with only the minimum features, allowing a team to collect the highest level of validated learning about customers with the lowest effort (Ries, 2009).

An MVP allows the project team to give valuable insights into how the product is performing early and make improvements with each following iteration. Interviewee 6 presented this advantage as follows: "*I prefer small loops. Okay, we have a roadmap to develop this massive project, but let us try to get out one core feature or an MVP. Get that in the hands of the client so that they can test it and see if they like it. Then we can iterate on that".*

If customers continuously see deliverables and provide feedback, they will likely be satisfied with the end product. Thus, frequent deliverables are valuable since customer satisfaction is a success criterion identified in the interviews. "*Delivery speed, quality*

of work, how many physical deliverables they receive at the end of the day affect your customer's satisfaction" (2).

5.2.5.7 Proper work handover

According to the interview findings, proper work handover is critical to the success of Agile projects; however, it is one of the less influential factors in Agile software development, ranked in the 21st position.

When team members hand over work to each other, some interviewees recognised how this can impact a project's success. Several times in an Agile project, work must be handed over from one person or team to another. Interviewee 1 details these instances: *"The design team would design and then hand over to the client, and the client hands over the back to us like a sign-off, then we hand over to the developers."*

They further explain that any miscommunication that happens in between affects project success. *"So, in each of those handover processes, if we end up with different teams or people, if there is a lack of communication, irregular translation or just miscommunication in general, then that would affect project success"* (1).

Thus, this work handover needs to be done correctly, as expressed by Interviewee 11: *"Handing over of sections to different team members need to be managed correctly, like having a proper handover so that the person leaving gives all of the information that they have, and the person being on-board has all of the information they need, and they do not have to ask many people."*

Some ways of ensuring the handover is done correctly according to the findings are to set up a meeting where the handover is done or have the handover documentation. Interviewee 7 describes how a handover meeting would be beneficial: "*Have a one-on-one meeting and go through the full flow. Something that might seem slightly more obvious to one person might not always seem as obvious to another, so there might be misunderstandings."* As they indicated, misunderstandings can be clarified if the handover is done in a meeting where the work is thoroughly explained and all the

relevant parties are present. There is little room for a person to misinterpret the work, or in this case, for the developers to misinterpret the designs.

Another way is to use handover documentation. Interview 2 advocates for handover documentation but only enough to provide the necessary information without being cumbersome such that people will not read it. *"There must be enough documentation of the project to allow effective handover while not being a limiting factor in the sense that you use valuable time and overspend time doing massive documents that no one is gonna read, but having enough documentation so you can clear the project to someone else and say the everything you need to know is in there*" (2).

Everyone can do the proper task without misunderstandings or misinterpretations when work is correctly handed over. Thus, the team will be more aligned with each other and be on track to complete the project without leaving out anything within scope.

5.2.6 Technical Factors

The technical factors category covers the tools, technologies, or techniques used in the project (Arcos-Medina and Mauricio, 2020, Chow and Cao, 2008, Hummel and Epp, 2015, Misra et al., 2006, Misra et al., 2009, Shakya and Shakya, 2020). It groups factors related to the engineering process of the software under development (Arcos-Medina and Mauricio, 2020). The technical factors identified in this study were suitable tools and technologies and adequate testing.

5.2.6.1 Suitable tools and technologies

Using suitable tools and technologies is the highest-ranked technical factor, with an overall ranking of 10th.

Selecting suitable tools and technologies entails considering the project requirements, client specifications, and team expertise. *"We consider all sorts of facts from beginning to end. How or which platforms will be used, and from that, identify which software would most accommodate the requirements."* (12).

It was also found that the tools and technologies must be decided before project commencement: *"When it comes to technologies for a project, they should always be discussed beforehand"* (7). The tools and technologies can be for communication, design, analysis, development, project management, integration, or testing. *"So, we need to decide before we even start a project. What tools will be used for communication, what tools will be used for reporting, and what other tools should be updated and when they should be updated." (4)*

The team members must be given access once the tools and technologies are determined to ease project commencement. "*If you have what you need, like the resources you need, like if you already have the GitHub access, you know, it is really easy to get started*." (3). Different team members might also require access to different tools and technologies: *"For the designer, it would be access to Figma, sketch, or what needs to be used in the specific client contexts." (1).*

The interviews also stated that for technology to be suitable, the project team must also be able to use it, thus familiar: *"I think it is just good to select technologies that are tested, and people are familiar with" (10).* Some existing literature cited using advanced tools and technologies (Azhar and Abdullah, 2022, Misra et al., 2009). However, in this study, the interviewees highlighted choosing familiar software, not necessarily the latest technology. *"Sometimes the latest is not the best, but having access to tools that your employees know how to use." (2)*

If the technology that needs to be used on a project is not familiar to the project team, then they need to be trained to use it, *"Yeah, that is why the equipment is important. Technical literacy should be equal to the equipment. So, let us say you are working on a cloud-based project. We should all be at that level. Maybe we get cloud certification or learn fundamentals so everyone has equal knowledge of what we do. To be able to produce good work altogether" (9).*

Another option can be adding individuals with the required expertise to use the tools and technology to the project: *"If it is a tool that a client wants that we do not have, we*

usually get the licence and get that tool and then get familiar with the techniques. So, you know, if the tool cannot be used by the expertise of anybody in the company, then sometimes we hire people for that project" (1)

Using advanced tools and technologies was identified as a critical success factor in the SLR. Conversely, the agile practitioners in this study emphasised using suitable, familiar technologies over advanced technologies. Azhar and Abdullah (2022) state that advanced technology will significantly improve project success rates. However, our findings align with a similar but quantitative study conducted in South Africa by Chiyangwa (2017) that found using suitable technologies and tools positively affects agile project success.

5.2.6.2 Adequate Testing

The interviewees indicated that the amount of testing done in a project must be enough to identify any issues. *"When testing, it is important to get people that will properly test your product"* (14). The testing should be done throughout the project, not just at the end, explains interviewee 7, *"There should be rigorous testing and not just at the end. For a ticket to be done, it needs to be fully tested, automated and manual. To avoid regressions, you need a testing environment divided by the testing framework you use."*

The interviewees believe testing in an environment similar to where the software will be deployed is better than testing in local environments. *"It is just better to have as soon as somebody says the code is ready. It goes live, and they can test it live"* (6). Software tested by individuals on their devices can still produce issues when deployed in the live environment.

Interview 6 states that testing can be done more efficiently through automated pipelines: "*If you have automated pipelines, you push this code to a branch or wherever, and it deploys it into its relevant environment. You can test the code in its expected environment so that you do not see any errors like, " Oh, it works on my*

machine" when it does not work live. So yeah, test-driven development and automated pipelines are important regarding the tools" (6).

Adequate testing also involves the customer—the need to do user acceptance testing (UAT). In User Acceptance Testing, the user does manual testing, and the goal is to ensure customer satisfaction (Pandit and Tahiliani, 2015). *"User acceptance testing needs to be done for the client to sign off on the projects"* (11). Doing UAT throughout the project increases the chances of success by increasing product quality and customer satisfaction. *"What I like is giving the client access to the test server; a customer involved in his product is going to have a good product"* (13)

Hence, if adequate testing is done on a project, it improves the chances of producing quality software and attaining customer satisfaction.

5.2.6.3 Agile software development techniques

Two of the attributes of agile software development techniques were identified by the interviewees: only having necessary documentation and consistent coding standards. Some researchers, including Chow and Cao (2008) and Stankovic et al. (2013), identified these attributes and three others not in this study: seeking simple design, extensive refactoring activities and accurate integration testing. Thus, it is significant to note that this discussion does not include those three other attributes commonly associated with the critical success factor, Agile software development techniques.

According to the findings, only necessary documentation should be done in an Agile software development project. Interview 4 supposes that *"a little bit of documentation helps, but it needs to be lightweight because that documentation should help and not hinder" (4).*

Interview 2 also believes that *"There must be enough documentation of the project to allow effective handover while not being a limiting factor in the sense that you use valuable time and overspend time doing massive documents that no one is going to read*" (2). This statement echoes the concept stated in the Agile manifesto that in Agile,

we value documentation but do not squander reams of paper on tomes that are seldom used and never maintained (Fowler and Highsmith, 2001). The project team does not need to spend time on unnecessary documentation; thus, it is critical to only produce what will be read and used.

Coding standards are a collection of rules that developers must comply with, which specify how the code should be formatted, for example, the name conventions that should be used (Timperi, 2004). Interviewee 7 highlights the importance of having consistent coding standards:*" I think coding standards are very important. Working on a feature that needs to be integrated with another feature you might not have developed is very common. This process is more straightforward when we have a common standard throughout the project (7).* Having consistent coding standards allows developers to be able to integrate their work with the work of others easily, which can save the team time. This also applies when a new developer needs to be added to a team; it shortens the time needed for project onboarding, as explained by interviewee 13: *"If every project in your company has the same folder structure, the same way of building queries, the same repositories or services, if it is all the same, no matter what project you step into, you immediately have a base idea of what to do.*

Thus, when a project team uses agile software development techniques that work for them, meaning the right amount of documentation for the project and following a consistent standard, this can increase the chances of delivering a project on time.

5.2.7 Project Factors

This category includes the factors relating to how project activities are carried out (Arcos-Medina and Mauricio, 2020, Chow and Cao, 2008, Hummel and Epp, 2015, Misra et al., 2006, Misra et al., 2009, Shakya and Shakya, 2020). It comprises factors related to the nature of the project during its development (Arcos-Medina and Mauricio, 2020). The project factors identified in this study were clear scope and requirement, adequate budget, adaptive schedule and project priority.

5.2.7.1 Clear scope and requirements

Having a clear scope and requirements was the highest-ranked project factor, showing its importance in project-related aspects. For a project to succeed according to the identified success criteria, it must fulfil the project scope and requirements. In order to do so, a project team needs a good understanding of the scope and requirements. Thus, communicating the scope to all team members is critical to the success of an agile project. *"Everyone needs to be on the same page with the same goal. So, from the designers to the clients to the developers, they must understand that picture and dynamics in the same context. If one has more information but is not provided to the rest, it will cause a ripple effect of issues going further"* (12).

If the team does not have a complete picture of what they are working to achieve, it can pose challenges in attaining project success. Each team member will only focus on their part without knowing how it fits into the project. Interviewee 3 states that such a way of working can lead to a situation where *"everybody is fixated on the detail, and the bigger picture is lost*" (3).

Therefore, getting an overview of the whole project is necessary, even if someone is only required to work on a small piece. *"It is sometimes easy to feel like you are not part of a team like you need to do your little thing. For the dev side, for them to be able to work on this one little thing or component, have something as part of the process where they are first given a bit of that overview of what it is they are building, what is the full system"* (15).

Several interviewees emphasised that even developers must be given the full scope of the project as sometimes this is only communicated to management, analysts, or designers. Seemingly, they usually get tickets to work on without the full scope. One developer said, *"If the main focus is not provided to us, I believe no proper software system will be going out; there would be many problems in the development" (12).* Another developer shared a similar view: "*Yeah, we like to have the context of the whole project. Then we could be much more efficient." (13)*.

To clarify the scope, the project management, team leads, and team members must understand the requirements. *"…certain clarity of what are the requirements in terms of timeline, in terms of budget, in terms of actual functional requirements and non-functional requirements. How clear are we on that? I think it is really important from a project perspective."* (5). Understanding the scope and requirements helps the project team stay on track with the task at hand. *"I think a critical thing is understanding the requirements because it is very easy, especially in design, to go wild. So I think making sure you understand what they need makes it a lot easier to stick to what you have planned*" (15).

5.2.7.2 Adequate Budget

The interviewees indicated that having enough budget to cover all the resources required for a project is vital for success. Interviewee 2 says that they often ask, "*Is there funding, adequate funding available?"* when deciding how many developers to put on a project and according to interviewee 14, this applies to all roles: *"So, obviously, like they, they have to like to have a realistic budget for the project that they want. So that will determine how many people you can have in the team and how quickly it will be pushed out, so they have to have a realistic budget."*

The available budget affects the human resources that can be put into a project and how much time they can spend on it. Interviewee 1 describes how this affects the quality of the project outcome, saying, *"Some clients cannot finance larger projects, which tends to affect how much time we can spend on it, and the less time we can spend on it, the less refined we can get the product to be, which is an unfortunate truth"*.Thus, according to the interviewees, for a project to succeed, the project budget must be adequate to allow the required number of people to work on a project for a long enough period to produce quality software.

5.2.7.3 Adaptive schedule

An adaptive schedule was identified as a critical success factor of Agile software development. *"Timeline is important for a project. So be flexible. And so, when we think of a timeline, we should think ahead, like worst case scenario, best case scenario"* (9).

This interviewee and many others referred to the schedule as a timeline; however, the term schedule was used as this is commonly used in research in this field.

An adaptive schedule does not entail forgoing deadlines but rather adjusting the time allocated to specific tasks and adjusting what is delivered in an iteration. *"Delivering on time is important, but that does not necessarily mean delivering the whole project. So, we need to be agile or pragmatic in what we deliver, so if we notice that we will not be delivered on time, we need to adjust the scope of what we will be delivered in that iteration and communicate it to the client"* (4).

In their study, Chow and Cao (2008) found that their results implied that project managers might not have to place as much weight on considerations such as project nature, project type, or project schedule when deciding to go for Agile development methodologies. This study identified a project schedule-related factor: adaptive schedule. However, it was ranked 22nd among the 25 CSFs. Therefore, as Chow and Cao (2008) stated, it might not be the most important CSF for organisations to focus on; however, it is still a critical success factor of Agile Software development.

5.2.7.4 Project Priority

According to the interviewees, each project has a different priority for the customer and the organisation. *"Whether it is a success or not, I think it is also about how the project is regarded, does the business care about this project?"* (10).

In the case study organisation, the same customer can have several projects with them, some more important to the customer than others, and project team members are moved accordingly. *"If the client is not into the project or bought into it one of the projects, it tends to fall on the back burner. So, you get people moving to high-value projects and the lower-value projects taking the backstage"* (1).

The priority that the organisation and the client place on the project significantly impact the time and effort the team can put into a project. *"I would say priority level matters. Some projects that I have worked on were not a big priority. So, it did not get all the*

attention it needed, and then it was rushed because it became important later" (11). Thus, the project team must assign the appropriate priority to a project to ensure it does not fall behind schedule or become neglected.

Commented [RH10]: appropriate what? Double check and proof read carefully before submitting to the editor. I would say re-read your document three timeas (with one version in printed form, preferably, helps to spot issues using different media) after cleanup and corrections, thereafter you send to the editor…

1.3 PRIORITISATION OF THE IDENTIFIED CSF

The frequency-based ranking in section 4.6 presented the priority of the identified critical success factors. The factor with the highest frequency and thus deemed the factor with the most significant influence on agile project success is organisational culture. 10.3% of the mentions by interviewees were either on organisational culture or its attributes. Accordingly, it was also the highest-ranked organisational factor.

The factors with the second highest ranking were accurate planning and estimation and customer Involvement. Both factors had 10.0% mentions. Notably, the difference in the number of mentions for these second-ranked factors and the top factor, organisational culture, was only 0.3%; thus, accurate planning and estimation and customer Involvement are also deemed as having a largely significant influence on project success.

The factors with the lowest ranking were adaptive schedule, project priority, small team, and societal culture. Each of these factors had a percentage frequency of 1%. Two of these lowest-ranking factors are project factors: adaptative schedule and project priority. They were ranked 22nd and 23^{rd,} respectively. Additionally, no factor in the project category was in the top ten rankings. The clear scope and requirements factor was the highest-ranked project factor in the 11th position, followed by adequate budget in the 18th position. The low rank of most factors in the project category suggests that this category is one of the less influential categories on Agile project success.

The other factors at the bottom of the list were small team and societal culture, belonging to the team and organisational categories. Thus, the team and organisational categories had the lowest ranking factors but the highest number of factors in the top ten. The organisational category had six identified factors, three of

which were in the top ten: organisational culture, managing customer expectations and communication. These three factors were ranked first, fifth and sixth, respectively. The team category had three of the four identified team factors in the top ten: team morale, team capabilities, and a coherent, self-organising team. These three factors were ranked fourth, eighth and ninth, respectively. Thus, the team factor had the majority of its factors in the top ten, but the organisational factors were in higher positions than the team factors. Therefore, it can be argued that the organisational category has the most influence on agile project success alongside the team category.

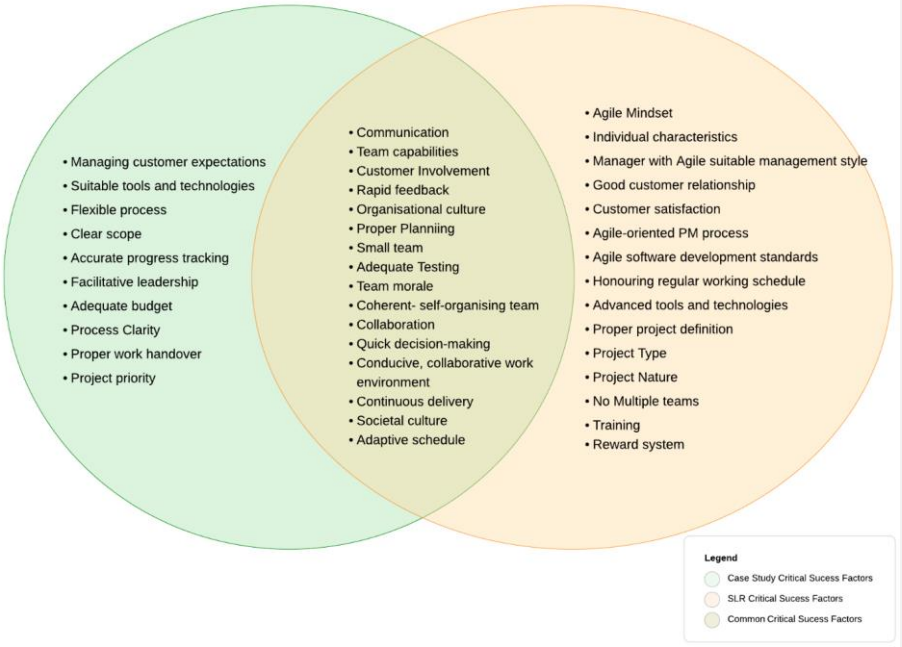
Planning and estimation was the highest-ranked factor in the process category in second place, followed by a flexible process in seventh place. The rest of the process factors were fairly distributed in the rankings; two of its lowest-ranked ones were in the bottom five: proper work handover and process clarity. Thus, it can be positioned that the process category as a whole has a moderate impact on agile project success compared to the other categories.

The highest-ranked technical factor was suitable tools and technologies. It was ranked in 10th place, and it was the only technical factor that was in the top ten of the ranking. The other two technical factors, adequate testing and agile software development techniques were ranked 15th and 19^{th,} respectively. Thus, the technical category does not have factors with the greatest nor the most negligible impact, similar to the process category.

5.3 BENCHMARKING THE FINDINGS WITH THE SLR FINDINGS

In Chapter 2, a systematic literature review was conducted, and the findings showed 33 factors identified from existing literature that spanned across findings from Asia, Africa, North America, South Africa and Europe. In this research, 25 critical success factors were identified from a case study of a single South African software development organisation. This section evaluates the similarities between the CSF identified in this study and the SLR. Figure 5.2 depicts the overlap of critical success factors in this study versus those identified in the SLR in a Venn diagram.

Figure 5.2: Venn diagram showing the overlap of critical success factors identified in this study versus those identified in the SLR.



Sixteen factors identified in this study were also identified in the SLR. The common factors in the organisational factors category were communication, organisational culture, conducive collaborative work environment, quick decision-making, and societal culture. In the team category, the common factors were team morale, coherent self-organising and a small team. There was only one common factor in the customer category: customer Involvement. The process category had three common factors: rapid feedback, proper planning, and continuous delivery. In the technical category, adequate testing was only one common factor. Lastly, there was only one common factor in the project category: an adaptive project schedule. The significant overlap suggests that this study's findings can benefit other organisations. Of the overlapping factors, organisational culture, accurate planning and estimation, customer Involvement, team morale, communication, team capabilities and coherent – self-organising teams are in the top ten. Thus, seven of the ten most highly ranked

factors that the agile practitioners mentioned are also considered critical in other organisations in different geographical regions.

The “suitable tools and technologies” factor can be considered similar to a factor in the systematic literature review “advanced tools and technologies”. However, based on the findings, the researcher deemed the factors to be different since, according to the interviewees, as discussed in section 5.2.6, an organisation choosing advanced tools and technologies or the latest technology might not always mean that the tools are suitable for the project.

5.4 CHAPTER SUMMARY

Commented [AG11]: Maybe chapter summary? Chapter conclusion

The section discussed the findings of the study. Twenty-five critical success factors of Agile software development in the South African software development industry were identified. A framework was generated based on this CSF. This framework included six categories: organisational, customer, team, process, technical and project, by which the CSF were grouped. The framework also included the criteria for project success given by the interviewees: time, cost, quality, scope schedule, customer satisfaction and adoption. The frequency-based ranking was discussed to highlight the observations made by the researcher.

6 CONCLUSION

Commented [RH12]: Remember, in the conclusion you cannot mention new information that has not been discussed in the dissertation. I.e., it must be mentioning (highlighting) what was already mentioned/discussed/provided…Double check

6.1 SUMMARY OF FINDINGS

The main research question was, *what are the critical success factors and their priority in Agile software development in the South African software development industry?* It was answered through the answering of the sub-research questions as detailed below:

Sub-research question **a)** *What are the critical factors in the success of Agile software development recognised by Agile practitioners in the South African software development industry?*

Commented [RH13]: The response to this section is lengthy, provide a summarized version of it…

Twenty-five critical success factors were identified. These factors were grouped into organisational, team, customer, process, technical and project categories. The

organisational category had six factors: Organisational culture, Managing customer expectations, communication, conducive-collaborative work environment, facilitative leadership, and societal culture. Managing customer expectations is presented as a new CSR or one that might particularly apply to South African Software development organisations. The agile practitioners emphasised that managing customer expectations before project commencement and throughout the project is critical to project success. Based on the findings, Agile practitioners should set realistic targets, communicate the process upfront, and proactively communicate any setbacks or changes in the project.

In the team category, four critical success factors were identified. The factors were team morale, team capabilities, coherent- self-organising team and a small team. These factors align with the following principle in the agile manifesto: developing projects with motivated team members, providing them with the work environment and reinforcement they need and trusting them to do the work (Fowler and Highsmith, 2001).

One customer factor was identified, which is customer involvement. The attributes identified for customer involvement differed from those initially identified by (Chow and Cao, 2008). This difference suggests that the aspects of customer involvement that affect Agile project success may differ in a South African context from those in other parts of the world.

In the process category, seven factors were identified as critical success factors. Accurate planning and estimation, flexible process, rapid feedback, continuous delivery, accurate progress tracking, process clarity and proper work handover. Three factors, flexible process, process clarity, accurate progress tracking, and proper work handover, were identified in this study but not in the literature review. The interviewees highlighted that if a process is communicated, the team members are more likely to follow all the procedures, thus working more cohesively and effectively. The interviewees emphasised that progress tracking needs to be done using a tool that allows it to be done accurately to influence a project positively. Then, with the proper

work handover factor, two distinct methods of doing so were highlighted: handover meetings or handover documentation containing only the necessary information.

Three factors were identified in the technical category: suitable tools and technologies, adequate testing, and agile software development techniques. The emphasis in the similar factor identified in the SLR was on using advanced tools and technologies. On the contrary, the agile practitioners in this study emphasised using suitable, familiar technologies over advanced technologies. With testing, the point stressed by the agile practitioners was having adequate testing, which is thorough and involves user acceptance testing.

Lastly, four critical success factors were identified in the project category: clear scope and requirements, adequate budget, adaptive schedule, and project priority. Adequate budget and project priority were project factors not identified in the SLR. The interviewees highlighted that an adequate budget to cover resources needed to undertake the project is vital for Agile project success, as a lack of resources can result in not meeting deadlines. Then, with project priority, the interviewees emphasised that an organisation needs to assign an appropriate project priority that communicates to the team the relevant importance and urgency of the project.

The answer above to sub-research question a) achieved the objective of identifying the critical factors in the success of Agile software development recognised by Agile practitioners in the South African software development industry.

Sub-research question **b)** *What is the priority of the identified critical success factors?*

The frequency-based ranking of the critical success factors was used to show the priority of the identified factors and infer the level of importance of their categories.

The critical success factor mentioned the most, along with its attributes, was organisational culture. The interviewees place the highest importance on organisational culture and its attributes: organisational structure, knowledge sharing, organisational buy-in and an organisation's understanding of Agile. The second-ranked

factors were accurate planning and estimation and customer Involvement, which belong to the process and customer categories, respectively. The difference between the highest-ranked and second-ranked factors was only 0.3%. Thus, all these three factors might need to be priorities by Agile practitioners seeking to improve project success rates.

The factors with the lowest ranking were adaptive schedule, project priority, small team, and societal culture; thus, it is poised that these are the factors with the least influence on agile project success. So, although they are critical and should be paid attention to when prioritising which factors to address to increase project success, agile practitioners may have to assign them a lower priority.

The team and organizational categories had the most factors in the top ten rankings. Thus, the organisational and team categories were believed to have the most significant influence on agile project success. Therefore, an organisation and project team must pay close attention to these two categories when aiming to improve Agile project success rates.

The process and technical factors were fairly distributed in the ranking, with both having only one factor in the top ten. Thus, these categories were deemed neither the most nor least influential on project success. So, when looking at these categories, it might be worth focusing on the highest-ranked factors, namely accurate planning and estimation and suitable tools and technologies.

None of the factors in the project category were ranked in the top ten, and two of the four project factors were in the bottom five: adaptative schedule and project priority. The low ranking of most factors in the project category suggests that this category is one of the less influential categories on Agile project success. Consequently, Agile practitioners may need to give the project category a lower priority when assessing which critical success factors category to address first.

The answer above to sub-research question b) achieved the objective of determining the priority of the identified critical success factors.

6.2 RESEARCH CONTRIBUTION

The study's main aim was to identify and establish a hierarchy of the critical success factors that affect Agile software development success as perceived by Agile practitioners in the South African software development industry. This objective was met along with the specific objectives listed in section 1.5, leading to the following vital contributions identified in this study:

- The research identified the critical factors in the success of Agile software development recognised by Agile practitioners in the South African software development industry.
- The research identified the attributes of the critical success factors that were applicable.
- The research determined the priority of the identified critical success factors that may assist practitioners in South African software development to focus on the more significant factors.
- The research presented what was poised as new and expanded critical success factors not identified in existing literature when conducting the SLR.
- The research may contribute to the scientific body of knowledge by presenting a framework for the critical success factors in agile software development in the South African software development Industry, thus achieving the objective of

Understanding the factors critical to an Agile project's success can help Agile practitioners focus on the essential factors. This can improve the success rates of software projects within the South African software development industry.

6.3 LIMITATIONS OF THE STUDY

The study was restricted to South African agile practitioners and hence limited to South African agile projects. On a global scale, the findings cannot be fully generalized. Only participants at the case study organisation were involved, so casual inferences cannot be made from the data as alternative explanations cannot be ruled out. The case study organisation was chosen because it had organisational members available and was

Commented [RH14]: Yes, very important…

Commented [AG15R14]: Mention it again???

willing to participate. Additionally, it was accessible to the researcher. A multi-case study can be done in future research.

The researcher sourced interviewees representing all their roles at the case study organisation. Still, not all the known roles in agile software development are represented. Thus, the findings are limited to the perceptions of the roles represented: software developer, project manager, UX designer, UX analyst, technical lead, tester and business analyst. Thus, future research can study all the known agile practitioner roles represented.

6.4 FUTURE RESEARCH

The findings of this study have motivated further investigation and testing of the proposed framework's effectiveness and success factors evaluation. The results provide additional information to researchers exploring methods to improve Agile software development in the South African software development industry. The case study organisation had some employees working fully remotely, others entirely in the office, and others working in both. Thus, it would be interesting to see if the critical success identified and their rankings differ for a South African organisation where the employees work remotely or in one location. This area of research will help develop a holistic view of Agile software development in the South African software development industry.

1.4 CONCLUDING REMARKS

The research conducted a case study using qualitative interviewees to identify the critical success factors of agile software development. During the analysis of the interviews, attributes of these factors were also identified and presented in the findings. A framework was developed that maps the critical success factors to the six key themes identified: organisational, team, customer, process, technical and project.

The framework includes some new and expanded CSFs based on the perception of South African agile practitioners: managing customer expectations, facilitative

leadership, flexible process, accurate progress tracking, process clarity, proper work handover and adequate budget. A frequency-based ranking of the priority of the identified critical success factors in agile software development was presented in a South African context. The ranking showed that, according to the agile practitioners in the case study, organisational culture is the CSF with the most significant influence on Agile project success. The team and organisational category had the most factors in the top 10 ranking. In contrast, the project category had the most low-ranked factors.

This study used qualitative interviewees; thus, the researcher could provide detailed perceptions from the Agile practitioners. Thus, the research can be an eye-opener to other Agile practitioners, particularly managers and leaders in Agile software development organisations. Research in this area was lacking in the South African context, and these findings provide information that can increase the probability of success of Agile software development projects in the South African software development industry.

7 REFERENCES

Agarwal, A., Gupta, S. & Choudhury, T. 2018.Published. Continuous and integrated software development using DevOps. 2018 International Conference on Advances in Computing and Communication Engineering (ICACCE), 2018. IEEE, 290-293.

Akbar, M. A., Mahmood, S., Shafiq, M., Alsanad, A., Alsanad, A. a.-A. & Gumaei, A. 2020. Identification and prioritization of DevOps success factors using fuzzy-AHP approach. *Soft Computing*.

Akhtar, A., Bakhtawar, B. & Akhtar, S. 2022. Extreme Programming Vs Scrum: A Comparison Of Agile Models. *International Journal of Technology, Innovation and Management (IJTIM),* 2**,** 80-96.

Al-Saqqa, S., Sawalha, S. & Abdelnabi, H. 2020. Agile Software Development: Methodologies and Trends. *International Journal of Interactive Mobile Technologies,* 14.

Alami, A., Krancher, O. & Paasivaara, M. 2022. The journey to technical excellence in agile software development. *Information and Software Technology***,** 106959.

Aldahmash, A., Gravell, A. M. & Howard, Y. 2017.Published. A Review on the Critical Success Factors of Agile Software Development. 2017 Cham. Springer International Publishing, 504-512.

Aldahmash, A., Gravell, A. M. & Howard, Y. 2018. Using Factor Analysis to Study the Critical Success Factors of Agile Software Development. *Journal of Software*

Aldahmash, A. M. 2018. *A review on the critical success factors of agile software development: an empirical study.* University of Southampton.

Altameem, E. A. 2015. Impact of agile methodology on software development. *Computer and Information Science,* 8**,** 9.

Commented [RH16]: Remove capital letters from the reference names included, see Amankwaa, L. (2016). CREATING PROTOCOLS FOR TRUSTWORTHINESS IN QUALITATIVE RESEARCH…Check throughout

Further, make sure they are properly formatted …For example, see "(2017, 2017//)" and "*http://assets. scrumtraininginstitute. com/downloads/1/scrumprimer121. pdf"*

Be pedantic with the cleanup and reworking of this document as you do not want to loose unnecessary marks…

Alvarez, A. & Sánchez, B. B. 2022.Published. Agile transformation processes: impact of factors in the success of failure: An ongoing work on identifying and characterising what influences transformations. 2022 17th Iberian Conference on Information Systems and Technologies (CISTI), 2022. IEEE, 1-4.

Amankwaa, L. 2016. CREATING PROTOCOLS FOR TRUSTWORTHINESS IN QUALITATIVE RESEARCH. *Journal of cultural diversity,* 23.

Ambler, S. W. 2009. The agile scaling model (ASM): adapting agile methods for complex environments. *Environments*, 1-35.

Anjani, R. F., Raharjo, T., Hardian, B. & Suhanto, A. 2021.Published. Measuring the relationship between organizational culture and project success: A survey of Agile software development teams. 2021 International Conference on Advanced Computer Science and Information Systems (ICACSIS), 23-25 Oct. 2021 2021. 1-9.

Arcos-Medina, G. & Mauricio, D. 2020. Identifying factors influencing on agile practices for software development. *Journal of Information and Organizational Sciences,* 44, 1-31.

Asnawi, A. L. 2012. *Investigating adoption of and success factors for agile software development in Malaysia.* University of Southampton.

Awad, M. 2005. A comparison between agile and traditional software development methodologies. *University of Western Australia,* 30, 1-69.

Azhar, N. A. & Abdullah, N. a. S. 2022.Published. Mapping Study of Critical Success Factor for Agile Software Project. 2022 IEEE International Conference on Computing (ICOCO), 2022. IEEE, 108-113.

Backstrom, C. H. & Hursh, G. D. 1963. *Survey Research*, Northwestern University Press.

Bandara, W., Miskon, S. & Fielt, E. 2011.Published. A systematic, tool-supported method for conducting literature reviews in information systems. ECIS 2011 proceedings [19th European conference on information systems], 2011. AIS Electronic Library (AISeL)/Association for Information Systems, 1-13.

Barki, H., Rivard, S. & Talbot, J. 1993. Toward an assessment of software development risk. *Journal of Management Information Systems,* 10**,** 203-225.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A. & Jeffries, R. 2001. Manifesto for agile software development.

Belassi, W. & Tukel, O. I. 1996. A new framework for determining critical success/failure factors in projects. *International journal of project management,* 14**,** 141-151.

Benbasat, I., Goldstein, D. K. & Mead, M. 1987. The case research strategy in studies of information systems. *MIS Quarterly***,** 369-386.

Boddy, C. R. 2016. Sample size for qualitative research. *Qualitative Market Research: An International Journal*.

Boehm, B. 2002. Get ready for agile methods with care. *Computer,* 35**,** 64-69.

Bogopa, M. E. & Marnewick, C. 2022. Critical success factors in software development projects. *South African Computer Journal,* 34**,** 1-34.

Bowen, G. A. 2003. *Social funds as a strategy for poverty reduction in Jamaica: An exploratory study.* Florida International University.

Bowen, G. A. 2005. Preparing a qualitative research-based dissertation: Lessons learned. *The qualitative report,* 10**,** 208-222.

Boyatzis, R. E. 1998. *Transforming qualitative information: Thematic analysis and code development,* Thousand Oaks, CA, US, Sage Publications, Inc.

Braun, V. & Clarke, V. 2006. Using thematic analysis in psychology. *Qualitative research in psychology,* 3**,** 77-101.

Braun, V. & Clarke, V. 2012. *Thematic analysis*, American Psychological Association.

Budgen, D. & Brereton, P. 2006.Published. Performing systematic literature reviews in software engineering. Proceedings of the 28th international conference on Software engineering, 2006. 1051-1052.

Bullen, C. V. & Rockart, J. F. 1981. A primer on critical success factors.

Campanelli, A. S., Bassi, D. & Parreiras, F. S. 2017.Published. Agile Transformation Success Factors: A Practitioner's Survey. *In:* Dubois, E. & Pohl, K., eds. Advanced Information Systems Engineering, 2017// 2017 Cham. Springer International Publishing, 364-379.

Cavaye, A. L. 1996. Case study research: a multi-faceted research approach for IS. *Information Systems Journal,* 6**,** 227-242.

Chiyangwa, T. B. 2017. Modelling the critical success factors of agile software development projects in South Africa.

Chiyangwa, T. B. & Mnkandla, E. 2017. Modelling the critical success factors of agile software development projects in South Africa. *South African Journal of Information Management,* 19**,** 1-8.

Chow, T. & Cao, D.-B. 2008. A survey study of critical success factors in agile software projects. *Journal of Systems and Software,* 81**,** 961-971.

Collins, D. 2003. Pretesting survey instruments: an overview of cognitive methods. *Quality of life research,* 12**,** 229-238.

Connelly, L. M. 2016. Trustworthiness in qualitative research. *Medsurg Nursing,* 25**,** 435.

Creswell, J. W. 2009. Research designs: Qualitative, quantitative, and mixed methods approaches. *Callifornia: Sage*.

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

Cucolaş, A.-A. & Russo, D. 2023. The impact of working from home on the success of scrum projects: A multi-method study. *Journal of Systems and Software,* 197**,** 111562.

Da Silva, K. M. B. & Dos Santos, S. C. 2015.Published. Critical factors in agile software projects according to people, process and technology perspective. 2015 6th Brazilian Workshop on Agile Methods (WBMA), 2015. IEEE, 48-54.

Dany, L., Urdapilleta, I. & Lo Monaco, G. 2015. Free associations and social representations: some reflections on rank-frequency and importance-frequency methods. *Quality & Quantity,* 49**,** 489-507.

Darke, P., Shanks, G. & Broadbent, M. 1998. Successfully completing case study research: combining rigour, relevance and pragmatism. *Information Systems Journal,* 8**,** 273-289.

De Mello, R., Uchôa, A., Oliveira, R., Oizumi, W., Souza, J., Mendes, K., Oliveira, D., Fonseca, B. & Garcia, A. 2019.Published. Do research and practice of code smell identification walk together? A social representations analysis. 2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), 2019. IEEE, 1-6.

Deemer, P., Benefield, G., Larman, C. & Vodde, B. 2010. The scrum primer. *Scrum Primer is an in-depth introduction to the theory and practice of Scrum, albeit primarily from a software development perspective, available at:http://assets.scrumtraininginstitute.com/downloads/1/scrumprimer121.pdf,* 1285931497**,**15.

Dieteren, J. 2022. *Agile project management beyond the software industry: An analysis of success factors and the influence on project performance.* University Innsbruck.

Dikert, K., Paasivaara, M. & Lassenius, C. 2016. Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software,* 119**,** 87-108.

Dörnyei, Z. 2007. *Research methods in applied linguistics: quantitative, qualitative, and mixed methodologies,* Oxford; New York, N.Y., Oxford University Press.

Dubé, L. & Paré, G. 2003. Rigor in information systems positivist case research: current practices, trends, and recommendations. *MIS Quarterly***,** 597-636.

Dudhat, A. & Abbasi, M. A. 2021. Discussion of Agile Software Development Methodology and its Relevance to Software Engineering. *ADI Journal on Recent Innovation,* 3**,** 105-114.

Dwivedi, N., Katiyar, D. & Goel, G. 2022. A Comparative Study of Various Software Development Life Cycle (SDLC) Models. *International Journal of Research in Engineering, Science and Management,* 5**,** 141-144.

Dyba, T. & Dingsoyr, T. 2009. What do we know about agile software development? *IEEE Software,* 26**,** 6-9.

Eisner, E. W. 2017. *The enlightened eye: Qualitative inquiry and the enhancement of educational practice*, Teachers College Press.

Elo, S., Kääriäinen, M., Kanste, O., Pölkki, T., Utriainen, K. & Kyngäs, H. 2014. Qualitative content analysis: A focus on trustworthiness. *SAGE open,* 4**,** 2158244014522633.

Etikan, I. & Bala, K. 2017. Sampling and sampling methods. *Biometrics & Biostatistics International Journal,* 5**,** 00149.

Fielding, N. G., Fielding, J. L. & Fielding, J. 1986. Linking data (Vol. 4). Beverly Hills. *CA: Sage,* 10**,** 9781412984775.

Fowler, M. & Highsmith, J. 2001. The Agile Manifesto. *Software development,* 9**,** 28-35.

Gaborov, M., Karuović, D., Kavalić, M., Radosav, D., Milosavljev, D., Stanisaljev, S. & Bushati, J. 2021. Comparative analysis of agile and traditional methodologies in IT project management. *Journal of Applied Technical and Educational Sciences,* 11**,** 1-24.

Gasson, S. 2004. Rigour in grounded theory research: An interpretive perspective on generating theory from qualitative field studies. *The handbook of information systems research.* IGI Global.

Ghayyur, S. a. K., Ahmed, S., Ali, M., Razzaq, A., Ahmed, N. & Naseem, A. 2018. A systematic literature review of success factors and barriers of agile software development. *International Journal of Advanced Computer Science and Applications,* 9**,** 278-291.

Goddard, W. & Melville, S. 2004. *Research methodology: An introduction*, Juta and Company Ltd.

Grünbaum, N. N. 2007. Identification of ambiguity in the case study research typology: what is a unit of analysis? *Qualitative Market Research: an international journal,* 10**,** 78-97.

Hamdani, M. & Butt, W. H. 2017.Published. Success and Failure Factors in Agile Development. 2017 International Conference on Computational Science and Computational Intelligence (CSCI), 14-16 Dec. 2017 2017. 981-986.

Hanslo, R., Vahed, A. & Mnkandla, E. 2019. Quantitative analysis of the scrum framework. *Advances in agile and user-centred software engineering.* Springer.

Heale, R. & Forbes, D. 2013. Understanding triangulation in research. *Evidence-based nursing,* 16**,** 98-98.

Heeager, L. & Nielsen, P. A. 2013.Published. Agile software development and the barriers to transfer of knowledge: an interpretive case study. Scandinavian conference on information systems, 2013. Springer, 18-39.

Hesse-Biber, S. & Leavy, P. 2011. The Practice of Qualitative Research. The Practice of Qualitative Research. Thousand Oaks.

Hoda, R., Salleh, N. & Grundy, J. 2018. The rise and evolution of agile software development. *IEEE Software,* 35**,** 58-63.

Hummel, M. & Epp, A. 2015.Published. Success factors of agile information systems development: A qualitative study. 2015 48th Hawaii International Conference on System Sciences, 2015. IEEE, 5045-5054.

Hussein, A. 2009. The use of triangulation in social sciences research. *Journal of Comparative Social Work,* 4**,** 106-117.

Ishak, N. M. & Abu Bakar, A. Y. 2014. Developing Sampling Frame for Case Study: Challenges and Conditions. *World journal of education,* 4**,** 29-35.

Javani, B., Thabe, M. & Dewa, M. 2022.Published. Cultural difference and IT project success: a fallacy or reality on the South African ICT sector. 33rd Annual Southern African Institute of Industrial Engineering Conference, 2022. Southern African Institute for Industrial Engineering.

Jiang, L. & Eberlein, A. 2008.Published. Towards a framework for understanding the relationships between classical software engineering and agile methodologies. Proceedings of the 2008 international workshop on Scrutinizing Agile practices or shoot-out at the Agile Corral, 2008. 9-14.

Jick, T. D. 1979. Mixing qualitative and quantitative methods: Triangulation in action. *Administrative Science Quarterly,* 24**,** 602-611.

Jintian, Y., Sukamani, D. & Kusi, M. 2022. Influence of Agile Leadership on Project Success; A Moderated Mediation Study on Construction Firms in Nepal. *Engineering Letters,* 30.

Johnson, J. 2018. *CHAOS report: decision latency theory: it is all about the interval*, Lulu. com.

Joseph, N., Marnewick, C. & Santana, M. 2016. *Agile software development and IT project performance in South Africa: a positive relationship?*

Joseph, N. M. C. & Santana, M. 2016.Published. Agile software development and IT project performance in South Africa: a positive relationship. The 25th International Conference for Management of Technology, 2016. 335-355.

Kamal, T., Zhang, Q., Akbar, M. A., Shafiq, M., Gumaei, A. & Alsanad, A. 2020. Identification and Prioritization of Agile Requirements Change Management Success Factors in the Domain of Global Software Development. *IEEE Access,* 8**,** 44714-44726.

Kaplan, B. & Maxwell, J. A. 2005. Qualitative research methods for evaluating computer information systems. *Evaluating the organizational impact of healthcare information systems.* Springer.

Kaplan, B. & Shaw, N. 2004. Future directions in evaluation research: people, organizational, and social issues. *Methods of information in medicine,* 43**,** 215-231.

Keele, S. 2007. Guidelines for performing systematic literature reviews in software engineering. Technical report, ver. 2.3 ebse technical report. ebse.

Kelle, E. V., Visser, J., Plaat, A. & Wijst, P. V. D. 2015.Published. An Empirical Study into Social Success Factors for Agile Software Development. 2015 IEEE/ACM 8th

International Workshop on Cooperative and Human Aspects of Software Engineering, 18-18 May 2015 2015. 77-80.

Khoza, L. T. & Marnewick, C. 2020. Waterfall and Agile information system project success rates, A South African perspective. *South African Computer Journal,* 32**,** 43-73.

Kitchenham, B. 2004. Procedures for performing systematic reviews. *Keele, UK, Keele University,* 33**,** 1-26.

Kouzari, E., Gerogiannis, V. C., Stamelos, I. & Kakarontzas, G. 2015.Published. Critical success factors and barriers for lightweight software process improvement in agile development: A literature review. 2015 10th International Joint Conference on Software Technologies (ICSOFT), 20-22 July 2015 2015. 1-9.

Kuhrmann, M., Tell, P., Hebig, R., Klunder, J. a.-C., Munch, J., Linssen, O., Pfahl, D., Felderer, M., Prause, C. & Macdonell, S. 2021. What makes agile software development agile? *IEEE Transactions on Software Engineering*.

Kulathunga, D. & Ratiyala, S. 2018. Key success factors of scrum software development methodology in Sri Lanka. *American Academic Scientific Research Journal for Engineering, Technology, and Sciences,* 45**,** 234-252.

Leau, Y. B., Loo, W. K., Tham, W. Y. & Tan, S. F. 2012.Published. Software development life cycle AGILE vs traditional approaches. International Conference on Information and Network Technology, 2012. 162-167.

Lehtonen, T., Suonsyrjä, S., Kilamo, T. & Mikkonen, T. 2015.Published. Defining metrics for continuous delivery and deployment pipeline. SPLST, 2015. 16-30.

Levy, Y. & Ellis, T. J. 2006. A systems approach to conduct an effective literature review in support of information systems research. *Informing Science,* 9.

Lincoln, Y. S. & Guba, E. G. 1985. *Naturalistic inquiry*, sage.

Linke, K. 2019.Published. Traditional and Agile Management Approaches. 12th ILERA European Congress, Düsseldorf, Deutschland, 2019.

Lukusa, L., Geeling, S., Lusinga, S. & Rivett, U. 2020. Teamwork and Project Success in Agile Software Development Methods: A Case Study in Higher Education. *Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality.* Salamanca, Spain: Association for Computing Machinery.

Majumdar, A. 2022. Thematic analysis in qualitative research. *Research Anthology on Innovative Research Methodologies and Utilization Across Multiple Disciplines.* IGI Global.

Malik, A., Aulawi, H. & Setiawan, R. 2019. Sistem Informasi Potensi Investasi Terintegrasi Menggunakan Scrum Agile. *Jurnal Algoritma,* 16**,** 108-119.

Marnewick, C. & Labuschagne, L. 2009. Factors that influence the outcome of information technology projects in South Africa: An empirical investigation. *Acta Commercii,* 9**,** 78-89.

Mason, J. & May, T. 2002. Qualitative Interviewing: Asking, Listening and Interpreting. *Qualitative Research in Action.* Sage Publications Ltd.

Mbelli, T. M. & Hira, J. J. 2016.Published. The perceptions of Agile methodology in South Africa. CS & IT Conference Proceedings, 2016. CS & IT Conference Proceedings.

Meenakshi, E., Singh, A. & Agrawal, D. 2020. A review on success factors of agile software development. *International Journal of Applied Science and Engineering,* 8**,** 33-36.

Miler, J. & Gaida, P. 2019.Published. On the agile mindset of an effective team–an industrial opinion survey. 2019 Federated Conference on Computer Science and Information Systems (fedcsis), 2019. IEEE, 841-849.

Mirnalini, K. & Raya, V. R. 2010.Published. Agile software development approach for quality software. 2010 International Conference on Educational and Information Technology, 2010. IEEE, V1-242-V1-244.

Misra, S. C., Kumar, V. & Kumar, U. 2006. Success Factors of Agile Software Development. *Software engineering research and practice,* 1**,** 233-239.

Misra, S. C., Kumar, V. & Kumar, U. 2009. Identifying some important success factors in adopting agile software development practices. *Journal of systems and software,* 82**,** 1869-1890.

Mkoba, E. & Marnewick, C. 2020. Conceptual framework for auditing agile projects. *IEEE Access,* 8**,** 126460-126476.

Mohajan, H. K. 2018. Qualitative research methodology in social sciences and related subjects. *Journal of Economic Development, Environment and People,* 7**,** 23-48.

Munassar, N. M. A. & Govardhan, A. 2010. A comparison between five models of software engineering. *International Journal of Computer Science Issues (IJCSI),* 7**,** 94.

Myers, M. D. 2019. *Qualitative research in business and management*, Sage.

Myers, M. D. & Avison, D. 2002. *Qualitative research in information systems: a reader*, Sage.

Myers, M. D. & Newman, M. 2007. The qualitative interview in IS research: Examining the craft. *Information and organization,* 17**,** 2-26.

Nascimento, L. D. C. N., Souza, T. V. D., Oliveira, I. C. D. S., Moraes, J. R. M. M. D., Aguiar, R. C. B. D. & Silva, L. F. D. 2018. Theoretical saturation in qualitative research: an experience report in interview with schoolchildren. *Revista brasileira de enfermagem,* 71**,** 228-233.

Nasehi, A. 2013. A quantitative study on critical success factors in agile software development projects; case study IT company. University of Borås/School of Business and IT.

Nguyen, D. S. 2016. Success factors that influence agile software development project success. *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS),* 17**,** 171-222.

Noteboom, C., Ofori, M., Sutrave, K. & El-Gayar, O. 2021.Published. Agile project management: a systematic literature review of adoption drivers and critical success factors. Proceedings of the 54th Hawaii International Conference on System Sciences, 2021. 6775.

Oates, B. J. 2006. *Researching Information Systems and Computing* London Sage Publications

Olorunshola, O. E. & Ogwueleka, F. N. 2022. Review of system development life cycle (SDLC) models for effective application delivery. *Information and Communication Technology for Competitive Strategies (ICTCS 2020).* Springer.

Overhage, S. & Schlauderer, S. 2012.Published. Investigating the long-term acceptance of agile methodologies: An empirical study of developer perceptions in scrum projects. 2012 45th Hawaii International Conference on System Sciences, 2012. IEEE, 5452-5461.

Pandit, P. & Tahiliani, S. 2015. AgileUAT: A framework for user acceptance testing based on user stories and acceptance criteria. *International Journal of Computer Applications,* 120.

Patton, M. Q. 1999. Enhancing the quality and credibility of qualitative analysis. *Health services research,* 34**,** 1189.

Pereira, J., Varajão, J. & Takagi, N. 2022. Evaluation of information systems project success–Insights from practitioners. *Information Systems Management,* 39**,** 138-155.

Perera, C. & Perera, I. 2019.Published. The Impact of Client Involvement towards Agile Project Success in Sri Lankan Software Industry. 2019 Moratuwa Engineering Research Conference (MERCon), 3-5 July 2019 2019. 279-284.

Perren, L. & Ram, M. 2004. Case-study method in small business and entrepreneurial research: mapping boundaries and perspectives. *International small business journal,* 22**,** 83-101.

Peters, Y. D. 2020. *Critical success factors that influence the performance of agile software development methodologies in organisations.*

Polit, D. & Beck, C. T. 2014. Essentials of nursing research. *Appraising evidence for nursing practice,* 8.

Presser, S., Couper, M. P., Lessler, J. T., Martin, E., Martin, J., Rothgeb, J. M. & Singer, E. 2004. Methods for testing and evaluating survey questions. *Methods for testing and evaluating survey questionnaires***,** 1-22.

Qatanani, D., Al-Tawara, F. & Qusef, A. 2021. Success factors of agile projects: case study for projects in Jordan during COVID-19 pandemic. *IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT).*

Radhakrishnan, A., Zaveri, J., David, D. & Davis, J. S. 2021. The impact of project team characteristics and client collaboration on project agility and project success: An empirical study. *European Management Journal*.

Riaz, M. N., Mahboob, A. & Buriro, A. 2018. Social success factors affecting implementation of agile software development methodologies in software industry of Pakistan: an empirical study. *International Journal of Advanced Computer Science and Applications,* 9**,** 94-98.

Ries, E. 2009. Minimum viable product: a guide. *Startup lessons learned* 3.

Rockart, J. F. & Crescenzi, A. D. 1984. Engaging top management in information technology. *Sloan Management Review (pre-1986),* 25**,** 3.

Rowlands, T., Waddell, N. & Mckenna, B. 2016. Are we there yet? A technique to determine theoretical saturation. *Journal of Computer Information Systems,* 56**,** 40-47.

Rubin, H. J. & Rubin, I. S. 2011. *Qualitative interviewing: The art of hearing data*, sage.

Russo, D. 2021. The Agile Success Model: A Mixed-methods Study of a Large-scale Agile Transformation. *ACM Trans. Softw. Eng. Methodol.,* 30**,** Article 52.

Salahshour Rad, M., Nilashi, M. & Mohamed Dahlan, H. 2018. Information technology adoption: a review of the literature and classification. *Universal Access in the Information Society,* 17**,** 361-390.

Salikhov, D., Succi, G. & Tormasov, A. 2020. An empirical analysis of success factors in the adaption of the scaled agile framework--first outcomes from an empirical study. *arXiv preprint arXiv:2012.11144*.

Saunders, M., Lewis, P. & Thornhill, A. 2007. Research methods. *Business Students 4th edition Pearson Education Limited, England*.

Shaikh, S. & Abro, S. 2019. Comparison of traditional & agile software development methodology: A short survey. *International Journal of Software Engineering and Computer Systems,* 5**,** 1-14.

Shakya, P. & Shakya, S. 2020. Critical success factor of agile methodology in software industry of Nepal. *Journal of Information Technology,* 2**,** 135-143.

Shameem, M., Khan, A. A., Hasan, M. G. & Akbar, M. A. 2020. Analytic hierarchy process-based prioritisation and taxonomy of success factors for scaling agile methods in global software development. *IET Software,* 14**,** 389-401.

Shameem, M., Kumar, C., Chandra, B. & Khan, A. A. 2017.Published. Systematic Review of Success Factors for Scaling Agile Methods in Global Software Development Environment: A Client-Vendor Perspective. 2017 24th Asia-Pacific Software Engineering Conference Workshops (APSECW), 4-8 Dec. 2017 2017. 17-24.

Shameem, M., Nadeem, M. & Zamani, A. T. 2023. Genetic algorithm-based probabilistic model for agile project success in global software development. *Applied Soft Computing*, 109998.

Shanks, G. & Bekmamedova, N. 2018. Chapter 7 - Case study research in information systems. *In:* Williamson, K. & Johanson, G. (eds.) *Research Methods (Second Edition).* Chandos Publishing.

Shehzad, N. & Kausar, M. 2021. Organizational Factors Impacting Agile Software Development Systematic Literature Review. *American Academic Scientific Research Journal for Engineering, Technology, and Sciences,* 84**,** 69-91.

Shenhar, A. J., Dvir, D., Levy, O. & Maltz, A. C. 2001. Project success: a multidimensional strategic concept. *Long range planning,* 34**,** 699-725.

Siau, K., Long, Y. & Ling, M. 2010. Toward a Unified Model of Information Systems Development Success. *J. Database Manag.,* 21**,** 80-101.

Simon, M. K. & Goes, J. 2013. Scope, limitations, and delimitations.

Snyder, H. 2019. Literature review as a research methodology: An overview and guidelines. *Journal of Business Research,* 104**,** 333-339.

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**,** 1663-1678.

Stelzmann, E., Kreiner, C., Spork, G., Messnarz, R. & Koenig, F. 2010.Published. Agility meets systems engineering: a catalogue of success factors from industry practice. Systems, Software and Services Process Improvement: 17th European Conference, EuroSPI 2010, Grenoble, France, September 1-3, 2010. Proceedings 17, 2010. Springer, 245-256.

Stoica, M., Mircea, M. & Ghilic-Micu, B. 2013. Software development: agile vs. traditional. *Informatica Economica,* 17.

Syeda, S. H. 2018. *An exploratory study to identify critical success factors of agile systems engineering.* The George Washington University.

Tam, C., Da Costa Moura, E. J., Oliveira, T. & Varajão, J. 2020. The factors influencing the success of on-going agile software development projects. *International Journal of Project Management,* 38**,** 165-176.

Tam, C., Moura, E. J. D. C., Oliveira, T. & Varajão, J. 2020. The factors influencing the success of on-going agile software development projects. *International Journal of Project Management,* 38**,** 165-176.

Thomas, J. 2008.Published. Introducing agile development practices from the middle. 15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2008), 2008. IEEE, 401-407.

Thummadi, B., Shiv, O., Berente, N. & Lyytinen, K. 2011.Published. Enacted software development routines based on waterfall and agile software methods: socio-technical event sequence study. International Conference on Design Science Research in Information Systems, 2011. Springer, 207-222.

Timperi, O. P. 2004.Published. An overview of quality assurance practices in agile methodologies. T-76.650 Seminar in Software Engineering, 2004. Citeseer.

Tona, C., Juárez-Ramírez, R., Jiménez, S., Durán, M. & Guerra-García, C. 2019.Published. Towards a Set of Factors to Identify the Success in Scrum Project

Delivery: A Systematic Literature Review. 2019 7th International Conference in Software Engineering Research and Innovation (CONISOFT), 2019. IEEE, 97-106.

Tsirakidis, P., Kobler, F. & Krcmar, H. 2009.Published. Identification of Success and Failure Factors of Two Agile Software Development Teams in an Open Source Organization. 2009 Fourth IEEE International Conference on Global Software Engineering, 13-16 July 2009 2009. 295-296.

Tyagi, S., Sibal, R. & Suri, B. 2022. Empirically developed framework for building trust in distributed agile teams. *Information and Software Technology,* 145**,** 106828.

Varghese, A., Kolamban, S., Sherimon, V., Lacap, E. M., Ahmed, S. S., Sreedhar, J. P., Ahmed, S. R., Harthy, H. A. & Shuaily, H. S. A. 2022. Modelling Control Measure Score of COVID-19 Outbreak Using Fuzzy c-Means-Based Adaptive Neuro-Fuzzy Inference System. *Information and Communication Technology for Competitive Strategies (ICTCS 2020).* Springer.

Wagener, R. P. 2012. *Investigating critical success factors in agile systems development projects.* North-West University, Potchefstroom Campus.

Walsham, G. 1995. Interpretive case studies in IS research: nature and method. *European Journal of Information Systems,* 4**,** 74-81.

Webster, J. & Watson, R. T. 2002. Analyzing the past to prepare for the future: Writing a literature review. *MIS quarterly***,** xiii-xxiii.

Williams, C. 2007. Research methods. *Journal of Business & Economics Research (JBER),* 5.

Wolf, C., Fu, Y.-C., Smith, T. & Joye, D. 2016. The SAGE handbook of survey methodology. *The SAGE Handbook of Survey Methodology***,** 1-740.

Wolfswinkel, J. F., Furtmueller, E. & Wilderom, C. P. 2013. Using grounded theory as a method for rigorously reviewing literature. *European journal of information systems,* 22**,** 45-55.

Xiao, Y. & Watson, M. 2019. Guidance on conducting a systematic literature review. *Journal of planning education and research,* 39**,** 93-112.

Yin, R. K. 1994. Discovering the future of the case study. Method in evaluation research. *Evaluation practice,* 15**,** 283-290.

Yu, J. 2018.Published. Research process on software development model. IOP Conference Series: Materials Science and Engineering, 2018. IOP Publishing, 032045.

Żelechowska, D., Żyluk, N. & Urbański, M. 2020. Find out a new method to study abductive reasoning in empirical research—*International Journal of Qualitative Methods,* 19**,** 1609406920909674.

Zhou, Y., Zhang, H., Huang, X., Yang, S., Babar, M. A. & Tang, H. 2015.Published. Quality assessment of systematic reviews in software engineering: A tertiary study. Proceedings of the 19th international conference on evaluation and assessment in software engineering, 2015. 1-14.

APPENDIX A: Systematic Literature Review Protocol

Researcher: Ashley Gwangwadza	
Supervisor: Mr Ridewaan Hanslo	
Title of the project: Identification and prioritisation of success factors in agile software development in the South African software development industry.	
Research Question: What are the critical success factors and their priority in Agile software development in the South African software development industry?	
Problem:	Organisations and their environments change rapidly, as do critical success factors for software development (Siau et al., 2010). As a result, researchers must constantly work to identify the critical success factors that determine whether a project succeeds or fails. (Shehzad & Kausar, 2021). To aid in answering the research question A literature review is an essential component of academic research (Webster and Watson, 2002). This is because knowledge progress needs to be based on already completed work (Xiao and Watson, 2019). Thus, to comprehensively answer the research question, a systematic literature review needs to be done to understand the work that has already been done
Objectives:	Main Research Objective:

	This research aims to identify and establish a hierarchy of the critical success factors that affect Agile software development success as perceived by Agile practitioners in the South African software development industry. Specific Objectives of SLR: a) To identify factors that affect the success of Agile software development. b) To identify the critical factors in the success of Agile software development recognized by Agile practitioners in the South African software development industry.
What are the keywords that will guide the study?	Agile methodologies, software development projects, Critical success factors
What are the combinations of search strings?	("Agile software development" OR "Agile" OR “scrum”) AND ("critical success factors" OR "success factors" OR "success") AND ("projects" Or "South Africa" OR "software development" OR "software organisation")
What are the databases that you will consult?	IEEE Xplore Digital Library, Science Direct, ACM Digital Library, SpringerLink
What are the exclusion/inclusion criteria?	Inclusion criteria For a source to be included in the research, it had to meet the following criteria:

	• Papers that identify the factors that influence Agile software development • Papers that discuss Agile software development in a software development organisation • Papers that discussed factors that contribute to the success of Agile software development • Conference papers, journal articles, book chapters, thesis and dissertations were considered for review Exclusion criteria A source was excluded from the research if it met the following criteria: • Papers with no explicit discussion about Agile • Papers that do not provide information on factors contributing to Agile software development • Papers not written in English • Papers where the full text is not available to the researcher • Duplicate papers (the same papers taken from different databases) • Duplicate reports of the same study (the most complete version will be selected)

APPENDIX B: SLR Data Extraction Table

Title: IDENTIFICATION AND PRIORITISATION OF SUCCESS FACTORS IN AGILE SOFTWARE DEVELOPMENT IN THE SOUTH AFRICAN SOFTWARE DEVELOPMENT INDUSTRY

Research question: What are the critical success factors and their priority in Agile software development in the South African software development industry?

Articles included in Systematic Literature review:

	Citation	Paper Title	Publication Type	Year	Country	Continent	Database	Criteria/Concepts/Dimensions	Main criteria/concept/dimension
1	(Aldahmash et al., 2017)	A Review on the Critical Success Factors of Agile Software Development	Conference Proceedings	2017	United Kingdom	Europe	SpringerLink	Organisational culture, Management support, Team capability, Individual Characteristics, Customer Involvement, Project definition	Customer, Organisational, Team, Process
2	(Aldahmash et al., 2018)	Using Factor Analysis to Study the Critical Success Factors of Agile Software Development	Journal Article	2018	United Kingdom	Europe	Google Scholar	Training, Management support, Testing, Tools and Technology, Team skills	Customer, Technical, Organisational, Team
3	(Aldahmash, 2018)	A review on the critical success factors of agile software development: an empirical study	Thesis	2018	United Kingdom	Europe	SpringerLink	Training, Team capability	Organisational, Team, Process
4	(Alvarez & Sánchez, 2022)	Agile transformation processes: impact of factors in the success of failure: An ongoing work on identifying and characterising what influences transformations	Conference Proceedings	2022	Spain	Europe	IEEE Xplore Digital Library	Management support, Interpersonal skills, Agile Mindset, Effective communication, Training	Process, Organisational, Team
5	(Arcos-Medina & Mauricio, 2020),	Identifying factors Influencing agile practices for software development	Journal Article	2020	Peru	South America	Google Scholar	The reward system, Management support	Organisational, Process, Team
6	(Asnawi, 2012)	Investigating the adoption of and success factors for agile software development in Malaysia	Thesis	2012	Malaysia	Asia	Google Scholar	Agile Mindset, Societal culture, Customer Involvement, Organisational culture, Management support, Training, Team Commitment	Customer, Organisational, Team

7	(Azhar & Abdullah, 2022)	Mapping Study of Critical Success Factors for Agile Software Project	Conference Paper	2022	Malaysia	Asia	IEEE Xplore Digital Library	Agile-friendly work environment, Management support, Effective communication, Training, Team capability, Organisational culture, Customer Involvement, Project Type, Schedule, Project Nature, Project definition	Customer, Organisational, Process, Team
8	(Chiyangwa, 2017)	Modelling the critical success factors of agile software development projects in South Africa	Journal Article	2017	South Africa	Africa	Google Scholar	Agile-friendly work environment, Self-organizing team, Organisational culture, Collaboration, Team size, Effective communication, non-separate teams, Team size, Customer relationship, Agile-oriented project management process, Agile-oriented requirements management process, Project definition, Agile software development standards	Organisational, Team, Process
9	(Chow & Cao, 2008)	A survey study of critical success factors in agile software projects	Journal Article	2008	USA	North America	Science Direct	Organisational culture, Team capability, Customer Involvement, Management support, Team motivation, Team commitment, No multiple teams, Self-organizing team, Project Type, Schedule, Project Nature, Testing, Tools and Technology, Risk analysis	Technical, Customer, Organisational, Team, Project

10	(Cucolaș & Russo, 2023)	The impact of working from home on the success of scrum projects: A multi-method study	Journal Article	2023	Denmark	Europe	Science Direct	Agile-friendly work environment, Effective communication	Organisational, Team
11	(Dieteren, 2022)	Agile project management Beyond the software industry: An Analysis of success factors and the influence on project performance	Thesis	2022	Austria	Europe	Google Scholar	Organisational culture, Agile-friendly work environment, Management support, Training, Team motivation, Agile Mindset, Customer Involvement, Team size	Customer, Organisational, Team, Process
12	(Ghayyur et al., 2018)	A systematic literature review of success factors and barriers of agile software development	Journal Article	2018	Pakistan	Asia	Google Scholar	communication, Management support, Collaboration, Organisational culture, Training, Reward system, Team commitment, Team capability, Agile Mindset, Team size, Agile-oriented project management process, Rapid Feedback, Project Definition, Agile software development standards, Risk analysis	Organisational, Team, Process
13	(Hamdani & Butt, 2017)	Success and Failure Factors in Agile Development	conference proceeding	2017	USA	North America	IEEE Xplore Digital Library	Management support, Team size, Testing, Tools and Technology, Agile software development standards	The project, Organisational, Technical, Process
14	(Hummel & Epp, 2015)	Success Factors of Agile Information Systems Development: A Qualitative Study	conference proceeding	2015	USA	North America	IEEE Xplore Digital Library	Management support, Team motivation, Organisational culture, Team commitment, Agile-friendly work environment, Individual Characteristics	Organisational, Team, Process

15	(Jintian et al., 2022)	An Empirical Study into Social Success Factors for Agile Software Development	Journal Article	2022	Netherlands	Europe	ACM Digital Library	Team commitment, Self-organizing team, Team Motivation	Team
16	(Kelle et al., 2015)	Influence of Agile Leadership on Project Success: A Moderated Mediation Study on Construction Firms in Nepal	conference proceeding	2015	Nepal	Asia	Google Scholar	Collaboration	Organisational
17	(Kulathunga & Ratiyala, 2018)	Critical success factors of scrum software development methodology in Sri Lanka	Journal Article	2018	Sri Lanka	Asia	Google Scholar	Project Type, Risk analysis, Schedule, Project Nature, Agile-friendly work environment, Organisational culture, Management support, Individual Characteristics, Customer satisfaction, Agile-oriented project management process, Agile software development standards	Customer, Project, Organisational, Team
18	(Misra et al., 2006)	Success Factors of Agile Software Development	Journal Article	2006	Canada	North America	Google Scholar	Decision time, Organisational culture, Individual Characteristics, Team capability, Societal culture, Training	Organisational, Team
19	(Misra et al., 2009)	Identifying some essential success factors in adopting agile software development practices	Journal Article	2009	Canada	North America	Science Direct	Societal culture, Agile-friendly work environment, Customer satisfaction, Self-organizing team, Organisational culture, Decision time, Individual Characteristics	Customer, Organisational, Team
20	(Nguyen, 2016)	Success factors that influence agile software development project success	Journal Article	2016	USA	North America	Google Scholar	Agile-friendly work environment, Team capability, Training, Effective communication, Team commitment, Team	Organisational, Team

								motivation, Organisational culture, Reward system, Individual Characteristics, Project Type, Schedule, Project Nature	
21	(Noteboom et al., 2021),	Agile project management: A systematic literature review of adoption drivers and critical success factors	Conference Paper	2021	USA	North America	Google Scholar	Management support, Training, Team size, Effective communication, Team commitment, Team motivation, Team capability, Customer relationship, Project Type, Schedule, Project Nature	Organisational, Team
22	(Peters, 2020)	Critical success factors that influence the performance of agile software development methodologies in organisations	Thesis	2020	South Africa	Africa	Google Scholar	Effective communication, Agile-oriented project management process, Agile-oriented requirements management process, Management support, Team commitment, Training, Organisational culture, Reward system, Agile-friendly work environment, No multiple teams, Team size, Self-organizing team, Team motivation, Team capability, Individual Characteristics, Customer relationship, Project Type, Schedule, Project Nature, Testing, Tools and Technology, Risk analysis	Customer, Organisational, Team, Technical
23	(Qatanani et al., 2021)	Success factors of agile projects: case study for projects in Jordan during Covid-19 pandemic	Conference Paper	2021	Jordan	Asia	IEEE Xplore Digital Library	Agile-friendly work environment, Self-organizing team, Organisational culture,	Customer, Project, Organisational, Team

								Reward system, Management support, Customer Involvement, Organisational culture, Testing, Tools and Technology, Training, Self-organizing team, Customer relationship	
24	(Riaz et al., 2018)	Social success factors affecting implementation of agile software development methodologies in software industry of Pakistan: an empirical study	Journal Article	2018	Pakistan	Asia	Google Scholar	Agile-friendly work environment, Organisational culture	Organisational
25	(Russo, 2021)	The Agile Success Model: A Mixed-methods Study of a Large-scale Agile Transformation	Journal Article	2021	Denmark	Europe	ACM Digital Library	Management support, Team capability, Individual Characteristics, Self-organizing team, Self-organizing team	Process, Organisational, Team
26	(Shakya & Shakya, 2020)	Critical success factor of agile methodology in the software industry of Nepal	Journal Article	2021	Nepal	Asia	Google Scholar	Collaboration, Agile-friendly work environment, Agile-oriented project management process, Agile-oriented requirements management process, Project definition, Team capability, Management support, Training, Organisational culture, Team size, Customer satisfaction, Agile software development standards	Customer, Organisational, Team, Process
27	(Shameem et al., 2020)	Analytic hierarchy process-based prioritisation and taxonomy of success factors for scaling agile methods in global software development.	Journal Article	2020	China	Asia	Google Scholar	Organisational culture, Self-organizing team, Team size	Customer, Process, Organisational, Team

28	(Shameem et al., 2023)	Genetic algorithm-based probabilistic model for agile project success in global software development	Journal Article	2023	India	Asia	Science Direct	Team motivation, Effective communication, Organisational culture	Organisational, Team, Technical, Process
29	(Shehzad & Kausar, 2021)	Organisational Factors Impacting Agile Software Development-A Systematic Literature Review	Journal Article	2021	Pakistan	Asia	Google Scholar	Management support, Customer Involvement, Training, Effective communication, Collaboration, Organisational culture, Team capability, Agile Mindset, Reward system, Agile-friendly work environment, Team size, Agile software development standards	Process, Customer, Organisational, Team
30	(Stankovic et al., 2013)	A survey study of critical success factors in agile software projects in former Yugoslavia IT companies	Journal Article	2013	Former Yugoslavia	Europe	Science Direct	Agile-oriented project management process, Effective communication, Self-organizing team, Management support, Organisational culture, Team capability, Societal culture, Collaboration, Customer Involvement, Project Type, Schedule, Project Nature, Agile software development standards	Customer, Process, Organisational, Team, Project, Process
31	(Stelzmann et al., 2010)	Agility meets systems engineering: A catalogue of success factors from industry practice	conference proceeding	2010	Germany	Europe	SpringerLink	Self-organizing team, Testing, Tools and Technology, Agile software development standards	Team, Technical, Process, Project
32	(Tam, Moura, et al., 2020)	The factors influencing the success of ongoing agile software development projects	Journal Article	2020	Portugal	Europe	Science Direct	Societal culture, Customer Involvement, Individual Characteristics, Training, Team capability	Customer, Organisational, Team, Process

33	(Wagener, 2012)	Investigating critical success factors in agile systems development projects	Thesis	2012	South Africa	Africa	Google Scholar	Management support, Team commitment, Agile-friendly work environment, Organisational culture, Reward system, Effective communication, Testing, Tools and Technology, Organisational culture, Team motivation, Team capability, Team size, Customer Involvement, Agile software development standards	Customer, Organisational, Team, Technical, Process
34	(Anjani et al., 2021)	Measuring the relationship between organisational culture and project success: A survey of Agile software development teams	Conference Paper	2021	Indonesia	Asia	IEEE Xplore Digital Library	Hierarchical culture, team culture, development culture, rational culture	Organisational
35	(Kouzari et al., 2015)	Critical success factors and barriers for lightweight software process improvement in agile development: A literature review	Conference Paper	2015	Greece	Europe	IEEE Xplore Digital Library	Training, Commitment, Resources, Involvement, Experience. Monitoring, methodology, feedback, communication	Organisational, Team, Process
36	(Campanelli et al., 2017)	Agile Transformation Success Factors: A Practitioner's Survey	Conference Paper	2017	Germany	Europe	SpringerLink	Organisation, Team, Process, Tools	Customer, Organisational, Team, Technical, Process
37	(Shameem et al., 2017)	Systematic Review of Success Factors for Scaling Agile Methods in Global Software Development Environment: A Client-Vendor Perspective	Conference Paper	2017	India	Asia	IEEE Xplore Digital Library	Management commitments, conducting social events, Coordination, Effective customer Involvement, Effective requirements analysis, 3C (Communication, Coordination and Control), Effective	Customer, Organisational, Team, Technical, Process

								leadership, Small team size, Software, Methodology, Short iteration, Human Resources, Management, conducting training, Motivating the developers, Experienced developers, Creating self-organizing teams, Knowledge, Integration, Encouraging for knowledge sharing, Encouraging for project visibility. Rich technological infrastructure	
38	(Tsirakidis et al., 2009)	Identification of Success and Failure Factors of Two Agile Software Development Teams in an Open-Source Organisation	Conference Paper	2009	Ireland	Europe	IEEE Xplore Digital Library	(1) constant and synchronous communication, (2) consistency in methodological development approach, (3) geographical dispersion management through an extensive testing culture and (4) FLOSSD experience in accepting and handling the environmental limitations.	Organisational, Process
39	(Perera and Perera, 2019)	The Impact of Client Involvement towards Agile Project Success in the Sri Lankan Software Industry	Conference Paper	2019	Sri Lanka	Asia	IEEE Xplore Digital Library	Client Communication Level, Client collaboration, Client Influence, Client Industry Knowledge, Client Flexibility,	Customer

APPENDIX C: SLR Literature Matrix

The literature matrix shows the themes covered in each article included in the systematic literature review.

No	Citation	Organisational	Team	Customer	Process	Technical	Project
1	(Aldahmash et al., 2017)	x	x		x		
2	(Aldahmash et al., 2017)	x	x	x		x	
3	(Aldahmash, 2018)	x	x		x		
4	(Alvarez and Sánchez, 2022)	x	x		x		
5	(Arcos-Medina and Mauricio, 2020),	x	x		x		
6	(Asnawi, 2012)	x	x		x		
7	(Azhar and Abdullah, 2022)	x	x	x	x		
8	(Chiyangwa, 2017)	x	x		x		
9	(Chow and Cao, 2008)	x	x	x	x	x	x
10	(Cucolaş and Russo, 2023)	x	x				
11	(Dieteren, 2022)	x	x	x			
12	(Ghayyur et al., 2018)	x			x	x	x
13	(Hamdani and Butt, 2017)	x	x		x		
14	(Hummel and Epp, 2015)		x				
15	(Jintian et al., 2022)		x				
16	(Kelle et al., 2015)	x					
17	(Kulathunga and Ratiyala, 2018)	x	x	x			x
18	(Misra et al., 2006)	x	x				

19	(Misra et al., 2009)	x	x	x				
20	(Nguyen, 2016)	x	x					
21	(Noteboom et al., 2021),	x	x					
22	(Peters, 2020)	x	x					
23	(Qatanani et al., 2021)	x	x	x		x		
24	(Riaz et al., 2018)	x						
25	(Russo, 2021)	x	x		x			
26	(Shakya and Shakya, 2020)	x	x		x			
27	(Shameem et al., 2020)	x	x		x	x		
28	(Shameem et al., 2020)	x	x		x	x		
29	(Shehzad & Kausar, 2021)	x						
30	(Stankovic et al., 2013)	x	x		x	x		
31	(Stelzmann et al., 2010)		x		x	x	x	
32	(Tam, Moura, et al., 2020)	x	x	x	x			
33	(Wagener, 2012)	x	x	x	x	x		
34	(Anjani et al., 2021)	x						
35	(Kouzari, 2015)	x	x		x			
36	(Campanelli, 2017)	x	x	x	x	x		
37	(Shameem, 2017)	x	x	x	x	x		
38	(Tsirakidis, 2009)	x			x			
39	(Perera and Perera, 2019)			x				

APPENDIX D: Pre-testing Procedure and Feedback

Follow-up probes and think-aloud were used in the interviews. The follow-up probes are interviewer-driven, and thus, the researcher used some of the examples of cognitive probes given by (Collins, 2003, Presser et al., 2004, Wolf et al., 2016).

An example of the cognitive probes included during the interview was: "What does the term "organisational factors" mean to you?" "Can you tell me more about that factor?" "You seemed hesitant to answer, “What were you thinking?". Finally, after the interviews, the participants were asked to give any general feedback; these questions included the following:

- Was the questionnaire comprehensive?
- Were the questions clear?
- Are there any questions you expected we would ask and did not?
- Was the questionnaire too long, too short, or about right?
- Is there any other feedback or areas of improvement you would like to provide regarding the questionnaire?

The demographics of the pre-testers are indicated below.

Participant	Pre-test Method	Job title	Agile Years of experience	Age	No of projects
1	Cognitive interview	Senior Software Developer	7	Over 21	Over 10
2	Cognitive interview	Design and Analysis team lead	5	Over 21	Over 10
3	Cognitive interview	Junior Software Developer	2	Over 21	1 - 5
4	Mail in report	Lecturer	-	-	-
5	Mail in report	Lecturer	-	-	-

Pre-test feedback

Participant 1 Interview:

They thought most of the questions were clear except the last one. The interviewee mentioned that the *"last question could have been more specific on the project side of things."* They did not understand the term "nature of the project" used in the question. Additionally, they thought that question because they did not understand it, it might have been helpful to be provided with examples as shown in the think-out-loud snippet.

Interviewee Thinking out loud:

"I think maybe you could break down some examples of how I could think about it because for me, then I was just like, oh, I do not know anything about the nature of a project, you know."

Lastly, the interviewee could not provide a single answer on their job title as they mentioned their title varies per project and depends on the client's needs.

Participant 2 Interview:

Interviewee 3 thought all the questions were *"pretty clear and easy to understand".* However, they had an issue answering question 2.2 in section B: Which agile methodologies have you used in a project? They mentioned that it is difficult to say which methodologies they use as they use a mix of the different methodologies in a single project as mentioned in their thinking.

Interviewee Thinking out loud:

"You might not follow a specific Agile methodology to the tee …. It is not something that comes up often in a working environment exactly what Agile methodology is being implemented."

They also mentioned that it might be useful for the questions to be provided with the interview questions afterwards. They should have the opportunity to do a retrospective after the interview where they can add more information as they could

not think of everything on the spot. Their response to the debriefing question in the snippet below is evidence of that.

Debrief Question asked by the interviewer:
Is there any other feedback or areas of improvement you would like to provide regarding the questionnaire?
Interviewee Response:
"If you have worked on quite a couple of projects, then it is a little bit difficult to think of all of the factors, and it feels like I am missing something that might be vital, but it is a little bit difficult to think of on the spot. So, I do not know if there is space to include feedback after retrospection."

Participant 3 Interview:
The interviewee's only feedback was that they would have appreciated receiving the questions beforehand to review them and prepare.

Debrief Question asked by the interviewer:
Is there any other feedback or areas of improvement you would like to provide regarding the questionnaire?
Interviewee Response:
"Yeah, I think maybe, if you are allowed to, send the questions beforehand. If I had the questions beforehand, I might have been able to come up with a lot more than I did on the spot."

The interviewer also observed that the interviewee would sometimes lose their train of thought whilst answering. When the question context was repeated, they would remember what they wanted to say as if they associated certain words in the context with a critical success factor. Additionally, with their job title, they struggled to provide a single answer.

Participant 4 Feedback:

The academics sent their feedback to the research questions via email, as seen in the snippet below:

"I briefly reviewed the interview guide.
In my opinion, the interview guide is well crafted, given the study's objectives.
I wish the candidate all the best with the interviews and research."

There were no recommended changes from this participant.

Participant 5 Feedback:

A participant who is an academic sent their feedback to the research questions via email in the form of annotations in a Word document copy of the interview questions.

The feedback was on specific questions, with the main feedback being on In Section B, Questions 2.2, the participant mentioned the questions: Which agile methodologies have you used in a project? It might not be clear, and there may be a need to mention whether it is in their most recent or all projects they have been a part of.

Pre-test Discussion of Findings

Section A Questions: Screening Questions
There were no issues in understanding these questions; however, it took a while for respondents to retrieve the estimated number of projects they had been a part of. However, as mentioned earlier, interview participant 2 did think these questions should have been asked beforehand.

Section B Questions: Interviewees' Background
All participants had difficulty answering the section B questions on the interviewee's background. Thus, this section took longer than anticipated for two participants as they had to explain the answers they had given. All participants were unsure what to say as their job titles varied from project to project and depended on the client's

needs. Each provided at least two job titles in different agile software development projects.

For question 2.2 in section B, which reads: Which agile methodologies have you used in a project? Participant 1 mentioned that they do not follow a specific agile methodology but apply agile principles. All the interviewees found this question difficult to answer, with others mentioning that they did not follow a specific methodology after the interviewee asked some probing questions because of their hesitation. For question 2.3 regarding the team size on the current question, the interviewee asked them to give the average time size. The interviewer did so because they all mentioned they are currently in more than one project, and the time sizes are inconsistent throughout the project.

Section C Questions: Critical Success Factors Questions

These questions are the core of the study; therefore, the respondents needed to understand them, and they did. Three participants felt they had more answers to give than they did but could not think of them on the spot. Two participants mentioned that they would have liked to have received the questions beforehand or had the opportunity to provide more answers. One participant required me to repeat and rephrase the last question for them to understand. They did not understand the meaning of the "nature of the project" in the context of the question. One participant commented on the structure of this section. They mentioned they would have wanted to know all the categories to be asked about before answering the questions.

Pre-test Conclusion

All the participants said the questionnaire was comprehensive, and no additional questions could have been asked. Three participants thought the interview length was just right, whilst one thought the screening questions could have been removed or done before making the interview shorter. A few issues with the section B questions need to be addressed, as most participants had difficulty answering most of the questions in this section. The Section A and Section C questions had minor feedback that could be addressed. Overall, the participants understood the question concepts for the core questions of the questionnaire; they did so consistently and in the way the researcher intended.

APPENDIX E: Interview Structure

Identification and prioritisation of success factors in agile software development in the South African software development industry

Interview Questions:

1. **Section A: Screening Questions**
   1.1. Which age group do you fall under:
   - Under 18 **(Stop interview)**
   - 18 - 21
   - Over 21

   1.2. How many years of experience do you have in agile software development projects?
   - Less than 3 months **(Stop interview)**
   - Between 3 and 12 months
   - 1 - 2 years
   - 3 - 4 years
   - Over 4 years

   1.3. How many agile software development projects have you been a part of?
   - 0 projects **(Stop interview)**
   - 1 – 5 projects
   - 6 - 10 projects
   - Over 10 projects

2. Section B: Interviewees' Background

The following questions briefly describe your agile project background.

2.1. What is your job title within an Agile software development project?

2.2. How often do you come into the office to work?

- A few days a week
- Every day of the week
- Never, I work from home every day.

2.3. Which agile methodology did you use in your most recent project?

2.4. What was the team size on your most recent project?

2.5. From your point of view, what criteria need to be met for a project to be deemed a success?

3. Section C: Critical Success Factors Questions

The following questions are to get your perspective on the factors that you believe are critical to the success of an agile software development project.

The questions will be asked based on the following categories: organisational factors, team factors, customer factors, process factors, technical factors, and project factors.

3.1. In your experience, what organisational factors *(factors relating to the organisational structure, organisational environment and administrative climate of the company)* affect the success of an agile software development project?

3.2. In your experience, what team factors *(factors relating to the people who manage and execute the project)* affect the success of an agile software development project?

3.3. In your experience, what customer factors *(factors relating to the people who sponsor the project or will use the product)* affect the success of an agile software development project?

3.4. In your experience, what process factors *(factors relating to how project activities are carried out*) affect the success of an agile software development project?

3.5. In your experience, what technical factors (factors relating to the tools, technologies, or techniques used in the project) affect the success of an agile software development project?

3.6. In your experience, what project factors *(factors relating to the project parameters)* affect the success of an agile software development project?

3.7. Are there other factors you have not mentioned that you believe are critical to the success of agile software development projects?

APPENDIX F: Ethical Clearance

5 December 2022

Reference number: EBIT/236/2022

Miss AS Gwangwadza
Department: Informatics
University of Pretoria
Pretoria
0083

Dear Miss AS Gwangwadza,

FACULTY COMMITTEE FOR RESEARCH ETHICS AND INTEGRITY

Your recent application to the EBIT Research Ethics Committee refers.

Approval is granted for the application with reference number that appears above.

1. This means that the research project entitled "Identification and prioritisation of success factors in agile software development in the South African software development industry" has been approved as submitted. It is important to note what approval implies. This is expanded on in the points that follow.
2. This approval does not imply that the researcher, student or lecturer is relieved of any accountability in terms of the Code of Ethics for Scholarly Activities of the University of Pretoria, or the Policy and Procedures for Responsible Research of the University of Pretoria. These documents are available on the website of the EBIT Research Ethics Committee.
3. If action is taken beyond the approved application, approval is withdrawn automatically.
4. According to the regulations, any relevant problem arising from the study or research methodology as well as any amendments or changes, must be brought to the attention of the EBIT Research Ethics Office.
5. The Committee must be notified on completion of the project.

The Committee wishes you every success with the research project.

Prof K.-Y. Chan
Chair: Faculty Committee for Research Ethics and Integrity
FACULTY OF ENGINEERING, BUILT ENVIRONMENT AND INFORMATION TECHNOLOGY

APPENDIX G: Company Authorisation Letter

200
Witch-Hazel Avenue
Highveld Techno Park
Centurion
0169

RM Kruger
082 697 1523
rendani@monkeyandriver.com

J Brosens
079 461 4755
jacques@monkeyandriver.com

26 October 2022

LETTER OF AUTHORISATION TO CONDUCT RESEARCH AT MONKEY AND RIVER

Dear EBIT ethics committee:

The purpose of this letter is to inform you that I give Ashley Gwangwadza permission to conduct the research titled "*Identification and prioritisation of success factors in agile software development in the South African software development industry*" in partial fulfilment of the requirements for a degree MIT in Information Systems at the University of Pretoria.

The company acknowledges that we have reviewed the interview and questionnaire procedures presented by the researcher in the invitation. The company accepts the procedure and authorizes the research to proceed. The research may be commenced upon approval from the EBIT ethics committee.

Sincerely,

Rendani Kruger
Director at Monkey and River

Signature

26/10/22
Date

APPENDIX H: Letter of Invitation

Dear _____,

I am Ashley Gwangwadza, and I invite you to participate in my research study on the *Identification and prioritisation of success factors in Agile software development in the South African software development industry.* It is being conducted as part of my master's in information technology at the University of Pretoria. The interview will be completed in 25 minutes. I would greatly appreciate your participation in this research and look forward to learning about your experiences and gaining new insights.
Please read through the information below and complete the informed consent form if you wish to participate.

APPENDIX I: Informed Consent Form

Project Information

Research Title:

Identification and prioritisation of success factors in Agile software development in the South African software development industry

Commented [RH17]: Agile should be in capital and apply throughout.

The Purpose of the Study:

This study aims to identify and prioritise the critical success factors of Agile software development in the South African software development industry. Although several aspects influence Agile's success, some factors are significant to Agile's project success. It is poised that the hierarchy of the critical success factors will assist practitioners in focusing on the most significant factors.

Commented [RH18]:

Commented [RH19]: Consider rephrasing, e.g., "Although several factors contribute to Agile success, some are significant to Agile project success"…. Reflect….

Research Study Description

Project objectives

This research aims to identify and establish a hierarchy of the critical success factors that affect Agile software development success as perceived by Agile practitioners in the South African software development industry. The following research objectives were set to aid in achieving the primary research objective:

a) To identify the critical factors in the success of Agile software development perceived by Agile practitioners in the South African software development industry.
b) To determine the priority of the identified critical success factors.
c) To provide research literature that contributes to the scientific body of knowledge on Agile software development.

How you identified and why you are being invited to participate in this research:

You were identified as a potential participant because your organisation is a South African organisation developing software using Agile methods and has been selected as a case study for the research. You were invited to participate because this research would benefit greatly from your involvement and experience with Agile software development.

Voluntary nature of the study:

Your participation in this research study is voluntary, and you can withdraw at any time without penalty.

Compensation:

There will be no compensation furnished for your participation in this study.

Costs, time, and place of research participation:

There is no cost to participate in this study, and the interview will take 25 minutes. The interview can be conducted face-to-face or online on a video conferencing platform such as Google Meets or Zoom.

Research results feedback:

Participants may receive the research results via email from the researcher if they wish.

Risks:

You are not expected to experience any risks. All semi-structured interview questions are not compulsory, and you are free to stop the interview at any time for any reason they are not obligated to disclose. All information collected will be confidential, so no personal risk is linked with participating in the inquiry. The data collected from interviews is for research purposes only and does not pose a risk to you. The study does not engage any physical risk, and it is highly unlikely that participants will be psychologically affected,

Benefits

The research will be made available to you if you are interested. The results of this study may be useful to you by providing insight into the critical success factors of agile software development and their priority. These insights may apply to your work and benefit your effectiveness in agile software development projects.

The result of this study may be useful to your organisation and other agile software development organisations in South Africa. Understanding the priority of critical success factors of agile software development might help agile practitioners focus on the most significant factors. This could improve the success rates of agile software development projects in South Africa.

Confidentiality

Your participation will remain confidential. I do not wish to analyse data individually, and all the data will be transferred to a computer program to analyse the entire group. This means that you are assured of anonymity. All information will be kept confidential, and only the researcher and the research supervisor will have access to the raw data. The researcher will not use your data for any purposes outside this research project. Also, the researcher will not include your name or anything else that could distinguish you in any study reports.

Commented [RH20]: Correct this sentence…

Voice Recordings

The researcher will record the interview session so the data can be later transcribed and analysed. If you have any questions or concerns about the recordings, please contact me using the details below.

Storage of Your Data

All the data collected (transcriptions and recordings) from the interviews will be stored on a password-protected Google Drive. Access to the original data will be limited to the principal investigator, Ashley Gwangwadza and the supervisor, Mr Ridewaan Hanslo. The coded data and data for analysis will be stored in the University of Pretoria research data repository. All data stored on the mentioned platforms will be disposed of and destroyed ten years after the study's commencement in the manner

prescribed and defined by the University of Pretoria's Information Management policy.

Agreement to participate in this research:

If you agree to participate in the interview and to be voice recorded, please complete the informed consent form below. Please email the signed informed consent form to me at u16206186@tuks.co.za. You will be emailed a copy of the form, including the researcher's signature.

Concerns

If you have any concerns regarding the nature of this project, please get in touch with the research supervisor using the details in the next section. Regarding concerns about the interview, please contact me with the details in the next section.

Researcher Contact Details

Researcher name: **Miss Ashley Gwangwadza**
Email address: u16206186@tuks.co.za
Phone **+27 (0)72 862 1217**
Faculty of Engineering, Built Environment, and Information Technology
Department of Informatics

Supervisor name: **Mr Ridewaan Hanslo**
Email address: ridewaan.hanslo@up.ac.za
Phone **+27 (0)12 420 3798**
Faculty of Engineering, Built Environment, and Information Technology
Department of Informatics

Informed Consent

Commented [RH21]: This should be on a separate page…page break

2.1 I, _____hereby voluntarily grant my permission for participation in the project as explained to me by Ashley Gwangwadza

2.2 The nature, objective, possible safety, and health implications have been explained to me, and I understand them.

2.3 I understand my right to choose whether to participate in the project and that the information furnished will be handled confidentially. I am aware that the results of the investigation may be used for the purposes of publication.

2.4 Upon signature of this form, the participant will be provided with a copy.

Signed: _____ Date: _____

Witness: _____ Date: _____

Researcher: _____ Date: _____

APPENDIX J: List of identified codes in the thematic analysis

Code	Frequency	Code group
Customer buy-in	3	Customer
User involvement	4	Customer
Customer feedback	4	Customer
Direct Communication	11	Customer
Customer Involvement	8	Customer
Understanding the process	6	Organisational
Organisational structure	7	Organisational
Knowledge sharing	5	Organisational
Organisational buy-in	7	Organisational
Organisation's understanding of Agile	5	Organisational
Top management is open to suggestions	3	Organisational
Internal Communication	10	Organisational
External Communication	2	Organisational
Effective and efficient communication	4	Organisational
Language Barriers	2	Organisational
Proximity and accessibility	3	Organisational
Collaboration	3	Organisational
Conducive work environment	3	Organisational
Managing customer expectations	11	Organisational
Organisational culture	7	Organisational
Facilitative leadership	6	Organisational
Societal culture	1	Organisational
Aligning process to project requirements	3	Process
Team feedback	3	Process
Management feedback	3	Process
Reviewer feedback	3	Process
Suitable ticketing system	3	Process
Accurate time estimation	10	Process
Planning	8	Process
Setting clear responsibilities	4	Process
Comprehensive tickets	8	Process
Communicating the process	2	Process
Delivering a minimum viable product first	2	Process

Code	Frequency	Code group
Handover documentation	1	Process
Handover meeting	1	Process
Flexible process	10	Process
Rapid feedback	4	Process
Accurate progress tracking	3	Process
Process clarity	3	Process
Continuous delivery	5	Process
Proper work handover	3	Process
Clear project definition	1	Project
Communicating the scope	5	Project
Understanding project requirements	6	Project
Adequate budget	5	Project
Project priority	3	Project
Adaptive schedule	3	Project
Technical skills	6	Technical
Team member Expertise	6	Team
Project manager's experience	2	Team
Technical training	2	Team
Team cohesion	10	Team
Team coordination	3	Team
Small team	3	Team
Team morale	6	Team
Individual motivation	8	Team
A sense of ownership	6	Team
Balanced work distribution	6	Team
Using familiar technologies	5	Technical
Automated testing	1	Technical
User acceptance testing	2	Technical
Only necessary documentation	3	Technical
Consistent coding Standards	2	Technical
Suitable tools and technologies	8	Technical
Adequate Testing	5	Technical

APPENDIX K: Language Editing Certificate

Date: 24 August 2023

Dissertation Author: Ashley Gwangwadza

Dissertation Title:

IDENTIFICATION AND PRIORITISATION OF SUCCESS FACTORS IN AGILE SOFTWARE DEVELOPMENT IN THE SOUTH AFRICAN SOFTWARE DEVELOPMENT INDUSTRY

To Whom it May Concern:

I hereby confirm that the manuscript referred to above was proof-read for language, spelling and grammar and formatting, as per the supplied guidelines as furnished to me by the author.

The research content contained within the manuscript has not been altered in any way and remains as drafted by the author.

The author may accept or reject any of my comments or suggestions upon receipt of the manuscript – all comments and edits have been clearly tracked and are visible to the author.

Kind regards

DJ Horn

Email: diane.horn@outlook.com

APPENDIX L: The Agile principles

1. The top priority is customer satisfaction - this is achieved through the continuous delivery of valuable software.
2. Changes in requirements are always welcome- the Agile method embraces modification for the client's competitive advantage, even when they occur late in the development process.
3. Frequent delivery of working software – this is to reduce the cycle time between software deliveries.
4. Developers and business individuals work collectively daily during the project – this emphasises the need for commitment from the customer and shared responsibility for the software development project.
5. A supportive environment with motivated people – management needs to trust the project team to make decisions and incorporate motivated individuals.
6. Face-to-face interaction is the most efficient form of communication – this method of communication is preferred over heavy documentation; communication prioritises understanding.
7. Progress is mainly measured by working software –software integration and testing are done in short cycles to meet clear software milestones.
8. Sustainable development – this means users and sponsors must be able to sustain a continuous pace indefinitely.
9. Attention to good design and technical excellence – the project's design should be repeatedly improved throughout the project.
10. Simplicity is essential – the aim is to give individuals simple guidelines that encourage creativity.
11. Self-organising teams – the best designs, requirements and architectures stem from self-organising teams.
12. Team behaviour adjusted to be more effective – It is vital for teams to regularly reflect on how to become more effective and make the necessary changes.

Commented [DH22]: Incomplete sentence - necessary what?