Identification and Prioritization of Success Factors in Agile Software Development in the South African Software Development Industry

A. Gwangwadza, R. Hanslo (*Member, IEEE*)

***Index Terms*—Agile methodologies, software development projects, critical success factors, Scrum, case study.**

***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 hierarchy of importance of these factors. This study addresses this gap through a case study of a South African software development organization. The study identifies and prioritizes 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 organization to identify the critical success factors. Twenty-five critical success factors were identified and grouped into six categories: organizational, team, customer, process, technical and project. Based on the findings, the research proposed a framework for critical success factors in Agile software development in the South African software development industry. The factors were then ranked based on the combined frequency of mentions of the factor and its descriptive attributes. Organizational culture was the highest-ranked factor, placing the highest importance on this factor. Additionally, the organizational and team categories had the most highly ranked factors, while the project category had the least ranked factors. **The research findings provide information that, if used by Agile practitioners in South Africa, may increase the success of Agile software development projects in the South African software development industry.**

I. INTRODUCTION

Software is vital in all areas of the modern world; however, software development is far from being deemed a perfect process [6]. The software performs basic and sophisticated tasks everywhere nowadays [96]. However, there are still high rates of failure in software development projects in South Africa and globally [97]. Success in software development projects is challenging to attain [73]. The difficulty in obtaining success is demonstrated by the numerous software projects that are abandoned, delayed, or rejected [6]. According to [28], 64% of software projects were challenged or failed since they were not on time, on target (scope) and within budget (cost).

Companies are choosing Agile methods over traditional methods as they are trying to become more responsive to the dynamic and fast-paced business environments they face [102]. Organizations spend much money implementing Agile development processes, hoping the software projects will be finished on time at a lesser cost [6]. The Agile methodology is incorporated into software development to create business value, deliver working software regularly and improve quality [66].

Agile methods have been increasingly adopted as the software development methodology of choice; however, some researchers criticize it because there are as many success stories as failure stories [63]. [29] 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 [13], [14], [49], [72], [75], [13].

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 [25]. For example, [75] states that some of their findings contradict the findings of [44]. Both were survey-based quantitative studies, but [75] did not find personal characteristics and societal culture as critical success factors, although they were identified in [44].

Additionally, in existing research, factors identified in one country may not apply to another. For example, [72] did not confirm that all the factors identified by [14] can be considered critical success factors in the Yugoslavian IT industry. Another inconsistency in existing research is that factors identified in one country may not apply to another. For example, [72] did not confirm that all the factors identified by [14] can be considered critical success factors in the Yugoslavian IT industry. Research in this area is lacking in a South African Context. In South Africa, the failure rate of ICT projects in 2008 was 27% [40]. That same year, 10.8 billion rands were lost on ICT projects [39]. Many industries in South Africa, such as finance and banking, health, mining, and telecommunications, use Agile software development methods and practices [85]. Thus, assessing the critical success factors of Agile software development in South Africa may provide

A. Gwangwadza is a postgraduate from the Department of Informatics, University of Pretoria, Gauteng, South Africa. E-mail: ashleygwa@gmail.com.

R. Hanslo is with the Academy of Computer Science and Software Engineering, University of Johannesburg, Gauteng, South Africa. E-mail: ridewaanh@uj.ac.za.

insights to help Agile practitioners complete the development process more successfully.

The inconsistencies in the quantitative research findings imply a need for the critical success factors of Agile software development to be analyzed using different methods and in the context of different countries. [48] 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 [95], IT projects in South Africa organizations have the following failure rates: small organizations at 50.2%, medium-sized organizations at 64.2%, large organizations at 16.9%, and the overall failure rate for South African organizations at 53.7%. Thus, there is a clear need for improvement.

Additionally, according to [55], there is not enough research into the hierarchy of importance of the critical success factors that affect Agile projects. The lack of research in this area is also believed to be the case for South Africa. 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.

The research aims to provide software practitioners with information that can increase the probability of success in future Agile software development projects by answering the following.

Research Question: *In the South African software development industry, what are the critical success factors and their hierarchy of importance in Agile software development?*

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

Sub-Research Question 1: *What are the critical factors in the success of Agile software development recognised by Agile practitioners in the South African software development industry?*

Sub-Research Question 2: *What is the hierarchy of importance of the identified critical success factors?*

To begin, a literature review was conducted to give a comprehensive picture of the existing critical success factors.

II. Literature Review

An efficient and well-conducted review provides a solid framework for knowledge expansion and theory-building facilitation [88]. Thus, a literature review was conducted to provide a theoretical foundation for the research.

The literature review begins with background information on the Agile manifesto. This is followed by a list of the critical success factors identified in existing literature. Lastly theoretical framework used is presented.

A. The Agile methodologies

Agile approaches have established themselves as the cornerstone of the software engineering field, radically altering the conception, development, and delivery of software projects [103].

The Agile methodology implies the ability to survive and be successful in an environment of continuous change [14]. 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 [42]. Incorporating the Agile method with system development is expected to aid organizations in becoming more productive in a rapidly changing and competitive business environment [49]. Agile methods are sets of new approaches in software development that have numerous practices such as Scrum, Extreme Programming (XP), Dynamic System Development (DSDM), Lean Development (LD), Feature Driven Development (FDD) and Crystal [7].

Scrum is the most well-known Agile software development approach worldwide [53]. According to the research by Joseph et al. [85] Scrum is also the most used Agile methodology in South Africa, with 82.9% usage. Additionally, it is the main methodology used in the organization selected for the case study.

Agile software development aims to provide “agility” in software development by addressing the inflexibility innate to traditional methodologies and curbing the consequent negative impact [75]. Where agility is defined by [101] *“the software team's capability to efficiently and effectively respond to and incorporate user requirement changes during the project life cycle”*. Agile software development is built upon an Agile philosophy detailed in the Agile Manifesto [86].

B. The Agile Manifesto

Although the concept of agility was not new, it gained widespread acceptance when it was first presented in 2001 as the Agile Manifesto [68]. 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 in the Agile manifesto are shown in Table I [10].

TABLE I

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
Individuals and interaction	processes and tools

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 [41]. Below is a

summary of the principles detailed in the Agile Manifesto according to [10]:

- 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-organizing teams
- Team behaviour should be adjusted to be more effective.

This Agile Manifesto is the foundational idea for Agile methodologies [65].

C. Project Success in Agile Software Development

Over time, the methods for determining project success and how to obtain it have changed [75]. One of the more popular metrics today to evaluate project success is the Iron triangle [54]. The iron triangle says a project is successful if it is delivered on time and within scope and budget [6]. However, this definition may not always be applicable. Some projects have run behind schedule and over budget but have succeeded remarkably [4].

[71] 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 [75]. In a study of the factors influencing the success of ongoing Agile software development projects by [75], they also considered customer satisfaction as a measure of success, time, and cost. They stated that the recent definition of project success in software development has emphasized 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 [11].

[14] 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. [13] emphasize 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 [54].

Considering all the definitions and measures of success in Agile software development, the researchers decided that for this research, the definition of success will be based on time, cost, quality, scope, and customer satisfaction. Thus, these elements of project success and the definitions by [14] and [70] are summarized in Table II.

TABLE II

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

D. Critical Success Factors

The critical success factors theory was presented to find the vital areas an organization needs to focus on to achieve a specific goal [68]. Where critical success factors are defined as *"the limited number of areas in which satisfactory outcomes will guarantee successful competitive performance for the organization department or individual"* according to [12]. [14] were among the first researchers to identify critical success factors in Agile software development based on a survey of 109 Agile practitioners.

Earlier studies by [14], [43] and [44] grouped the themes into categories named Organizational, People, Process, Technical and Project. However, more recent papers [7], [25] and [66] divide the people factor into team and customer categories.

In interpretive research, it is preferable to maintain a high level of openness to the field data and a willingness to revise early theories [82]. Thus, it is poised that splitting the people factors into customer and team factors would be beneficial not to generalize “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. Thus, it was poised to use the categories catering to earlier and most recent literature, leading to the themes in Table III.

The identified CSFs were then grouped into these categories, and the critical success factors are displayed in Table IV.

TABLE III

CATEGORIES OF CRITICAL SUCCESS FACTORS

Category	Definition	Source
1. Organizational	This category covers factors relating to the organizational structure, organizational environment and administrative climate of the company.	[6], [13], [24], [41], [64]
Team	This category covers factors relating to the people who manage and execute the project.	[6], [24], [64]
3. Customer	This category covers factors relating to the people who sponsor the project or will use the product.	[6], [24], [64]
4. Process	This category covers factors relating to how project activities are carried out.	[6], [24], [64]
5. Technical	This category includes factors relating to the tools, technologies, or techniques used in the project.	[6], [13], [24], [41], [64]
6. Project	This category covers relating to the project parameters.	[6], [13], [24], [41], [64]

TABLE IV

CRITICAL SUCCESS FACTORS OF AGILE SOFTWARE DEVELOPMENT

Factor	Freq.	Source
Organizational		
Management support	19	[13], [4], [2], [7], [22], [6], [53], [33], [68], [49], [64], [70], [24], [54], [78], [61], [18], [8], [5]
Organizational culture	20	[13], [65], [4], [70], [7], [53], [22], [33], [56], [68], [42], [12], [24], [54], [78], [64], [65], [18], [8], [67]
Societal culture	5	[7], [73], [70], [42], [42]
Agile-friendly work environment	13	[53], [48], [33], [22], [56], [68], [12], [54], [24], [42], [78], [16], [8]
Decision time	4	[42], [42], [53], [68]
Collaboration	8	[71], [30], [13], [22], [4], [2], [68], [64]
Effective communication	12	[53], [48], [22], [68], [49], [12], [78], [65], [18], [8], [5], [71]
Training	15	[7], [53], [4], [48], [22], [68], [49], [64], [73], [42], [54], [18], [8], [5]
Reward system appropriate for Agile methodologies	7	[6], [53], [48], [22], [68], [54], [78]
Team		
Team motivation and commitment	11	[13], [65], [7], [53], [22], [78], [27], [48], [49], [24], [18]
Team capabilities	16	[13], [22], [2], [3], [41], [48], [49], [53], [54], [61], [64], [68], [70], [73], [78], [8]
Individual Characteristics	11	[48], [33], [73], [41], [2], [24], [42], [53], [61], [18], [5]
No multiple teams	3	[53], [12], [13]
Team size	12	[13], [65], [22], [53], [68], [49], [64], [12], [54] ,[24], [78], [18]
Agile Mindset	5	[7], [22], [68], [18], [5]
Coherent, Self-organizing team	11	[13], [65], [53], [54], [12], [61], [27], [71], [53], [70], [42]
Manager with Agile suitable management style	8	[53], [56], [68], [49], [54], [78], [61], [27]
Customer		
Customer Involvement	18	[13], [65], [22], [3], [2], [7], [53], [73], [48], [33], [64], [70], [24], [54], [42], [78], [67], [8]
Good customer relationship	5	[53], [49], [64], [12], [54]
Customer satisfaction	3	[33], [64], [42]
Process		
Agile-oriented project management process	10	[2], [70], [53], [48], [33], [22], [64], [12], [18], [8]
Agile-oriented requirement management process	7	[53], [48], [49], [64], [12], [65], [5]
Rapid feedback	4	[22], [24], [54], [67]
Proper project planning	6	[48], [64], [42], [2], [24], [78]
Proper project definition	8	[33], [22], [68], [49], [70], [12], [2], [8]
Risk analysis	4	[53], [33], [22], [13]
Agile software development standards	13	[4], [2], [70], [53], [33], [22], [68], [64], [42], [12], [24], [78], [71]
Continues Delivery strategy	8	[13], [4], [2], [70], [53], [33], [70], [71]
Honouring a regular working schedule	5	[13], [6], [48], [54], [78]
Technical		
Advanced tools and technologies	6	[48], [4], [68], [24], [65], [8]
Testing	6	[53], [48], [42], [78], [8], [71]
Project Factors		
Project Type	5	[53], [33], [49], [42], [8]
Project Nature	4	[70], [13], [53], [12]
Adaptive project schedule	5	[70], [13], [53], [48], [33]

E. Organizational Factors

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 organizational 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 organization and a dynamic and fast-altering firm [13], [43], [55] and [58].

[44]. 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 includes layout, pair programming accommodation, work location, and a high degree of agility were also identified in this category [17], [23], [35], [55].

The decision-time factor's attributes are short decision times and organizational 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 [2].

Knowledge sharing is identified as organizational factor that increases the chances of Agile practices being successful. The Training factor's 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 [2], [13]. The reward system factor has the following attributes: recognition of good work and incentive programs.

F. Team Factors

Motivated team members should be the foundation of Agile software development [2]. 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 [19], [28], [51]. 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 [73]. 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 [14], [72], [75], [80].

The individual characteristics include a collaborative attitude, honesty, trust, a sense of responsibility, eagerness to learn, and interpersonal and communication skills [57]. The team size factor refers to the project having a small team. Teams are often small to facilitate collaboration and communication [80]. The Agile mindset factor is attributed to understanding the Agile methodology and aligned personnel per the Agile manifesto [19].

A coherent, self-organizing team has a sense of ownership and team thinking and takes responsibility [28], [63]. 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 [51], [63].

G. Customer Factors

[2] states that customer involvement is crucial during Agile software development. It includes several attributes such as customer commitment, 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 [4], [7], [14], [69], [75].

A good customer relationship is healthy between the organization, project team, and customer [56]. Customer satisfaction means the customer requirements have been met, and the product produced is fit for use [44], [66].

H. Process Factors

The whole process lifecycle should be Agile for success in Agile software development, from requirements elicitation to management [24]. 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 [2], [19.] The Agile-oriented requirement management process means customers elicit and prioritize requirements [5].

The rapid feedback factor refers to having short iterations to deliver the output, and face-to-face feedback sessions [56], [2] 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 [43], [66]. This factor aligns with the proper project definition factor, which requires an accurate effort estimation and setting realistic estimations [35]. The risk analysis factor refers to analyzing 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 critical features first [3], [14], [55], [72]. The final factor in this category is honouring a regular working schedule without overtime.

I. Technical Factors

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

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 [2], [24], [69].

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 [8], [80]. 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 [13], [23].

J. 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 [51]. The nature of the project needs to be non-life-critical software, yet it may be corporate mission-critical software [72]. Lastly, an adaptive project schedule with dynamic and accelerated project schedules is more likely to succeed [55].

III. Theoretical Framework

Several researchers on the critical success factors in Agile software development do not employ a specific framework but rather synthesize existing frameworks and use that. Constructs from TAM, TRA, TPB, and UTAUT were used by [14] and [44] to identify the CSFs for Agile software development projects. In these studies, and several others [2], [3], [4], [13], [14], [49], [72], they categorized the critical success factors into five categories: organizational, people, process, technical and project.

However, other studies, such as those by [66] and [25], split the people factors into team and customer. [66] 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.

Therefore, considering the approach taken by other research and the fact that the research aims to discover findings in a country were a lot of this research does not exist a revised theoretical framework that encompasses aspects from the studies mentioned above based on the research model by [14] and the later revisions by [78] and [66] was constructed and used in this research. This Framework is shown in Fig. 1.

IV. Method

A. Overview

The variables that affect Agile software development are complex and, like other areas in IS, rely heavily on people to succeed [75]. Reference [37] emphasizes that researchers in the IS field should study the technical and social facets of IS. This type of research is suited to an interpretive approach. Employing the interpretive approach, the authors in this study will be able to increase the understanding of the critical

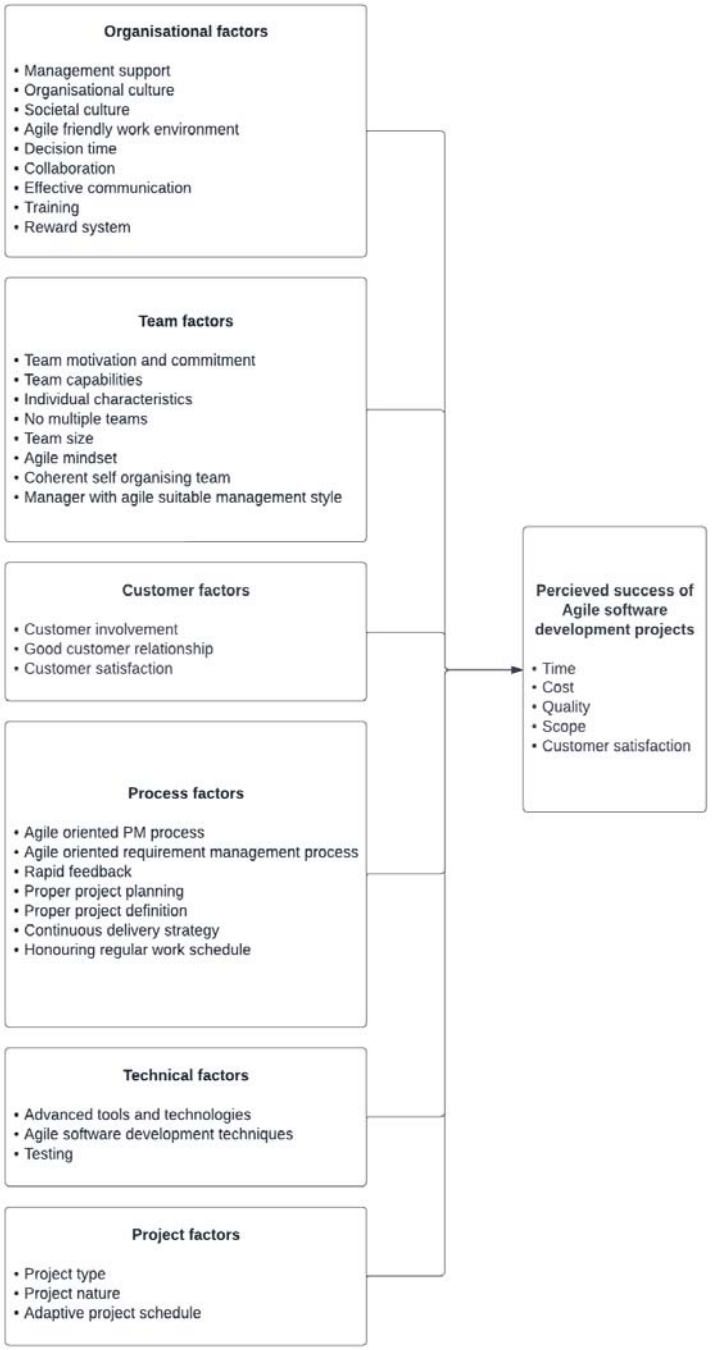

Fig. 1. Research Theoretical Framework.

success factors of Agile software development in South African software organizations. Thus, the interpretive research paradigm was deemed the most suitable for this

study.

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

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).

According to [89], there is distinctiveness in using the case study, which allows researchers to get comprehensive knowledge of a phenomenon in its context. Single case studies allow researchers to examine phenomena extensively and get an in-depth explanation and understanding [19]. Thus, a case study is an ideal strategy for a complex phenomenon that requires in-depth holistic inquiry [21].

For this study, the qualitative research method was selected. Qualitative research explores and gains insights into how people understand a situation or problem [15]. It gives the researcher holistic insights into deep, contextual data by allowing them to engage in conversation with research participants in a natural setting [27], [16], [41]. By doing qualitative research, the author interacted with the Agile practitioners in the South African software development industry to better understand the critical success factors in Agile software development.

B. Data Collection

One of qualitative research's most common and vital data collection tools is interviews [47]. Qualitative interviews in research act as night-vision goggles [45], allowing us to see what is not typically visible and to examine what is usually looked at but hardly ever seen [62]. 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 [90]. Triangulation in research is *"using more than one approach to researching a question"* [91]. Triangulation can refer to utilizing two or more theories, data sources, methods, or researchers [91]. Data source triangulation was used in this research. [31] define data triangulation as obtaining and analyzing data from multiple sources to get a holistic picture of the research topic. Three data triangulation techniques exist: time, space, and person [90]. Hence, the interviews in this study were conducted at different times (phases in software development projects), using people with different roles and years of experience.

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. The feedback from the pre-test then led to the interview structure in Appendix A. 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 interviews were carried out with the participants informed of the leading research goal without giving too much detail that could bias their opinions on the research issue. The interviews were recorded, transcribed, and coded using Atlas.ti, a qualitative research analysis tool. This process led to the identification and prioritization of the critical success factors.

C. Sampling

Non-probability sampling methods were used. Non-probability sampling is a technique that chooses a set of participants for a study using non-random methods [92]. The population of interest was Agile practitioners in South Africa, and a case study strategy was chosen for this research; thus, the sample frame was Agile practitioners at the selected South African software organization.

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

Convenience sampling selects the case to be studied based on how easily the researcher can reach it [35]. Thus, the selected organization for the case study met the following practical criteria listed by [20]: 1) easy accessibility, 2) geographical proximity, 3) availability at a given time, and 4) willingness of organizational members to participate.

Purposive sampling is a non-probability sampling method applied to choose a sample of individuals from a population [92]. It allows researchers to apply judgment when choosing participants [55]. In qualitative research, it is necessary to represent various voices by interviewing diverse people within an organization [46]. [47] argue that to overcome several biases, such as elite bias, researchers must involve various participants in their sample at different organizational levels. However, with a purposive sample, members are chosen based on their expertise, knowledge, and associations concerning the research subject [36]. Thus, the authors selected participants that would offer the most

diversity in terms of years of experience and role as provided by the company founders. This allowed for triangulation of participants and gain insights from a diverse range of them. Table V provides as overview of the interview participants.

[59] found that 95% theoretical saturation confidence was reached in 11 and 13 interviews. [59] illustrated that data saturation usually occurs at 12 interviews within a somewhat homogeneous population. The authors used this estimated initial number of 12 participants. Thus, 12 interviews were initially conducted. After the 12 interviews, a thematic analysis was performed. A few new labels (codes) were added in the 12th interview; thus, the interviewing was scheduled once more, thereafter conducting the thematic analysis. At 15 interviews, no new labels (codes) were identified in the last two interviews. Thus, it was determined that theoretical saturation was reached and ended at 15 interviews.

D. Data Analysis

The main objective of qualitative data analysis is to comprehend the pursuit of order and coherence [30]. In order to do so, a thematic analysis was used to analyze the data.

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 [30]. 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 themes and codes were refined into categories.

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 [98]. Frequency-based ranking is an approach taken by other researchers in this field that does not take a quantitative approach [8], [56] and [50].

The free association allows the researcher to process data produced directly from the free expression of individuals [98]. 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 method was used by some researchers in the IS field, such as [54], [48], [99], and [100]. The most cited factor(s) is ranked first, followed by the second most frequently mentioned factor, and so on [99]. In their ranking, [100] associated each descriptive answer with the definitions in their catalogue, resulting in evoked terms and associations. Similarly, the authors 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.

E. Case Setting

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 organization in the rest of the research, and interviewees will be referred to by numbers from 1 to 15.

The case study organization is a human-centred software organization that focuses on developing custom-designed systems using Agile software development practices. The case study organization 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 organization 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 organization as relatively flat. There are two owners/founders who also act as the organization'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 organization'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.

F. Characteristics of Interviewees

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 details of the interviewees is shown in Table V.

TABLE V

CHARACTERISTICS OF INTERVIEWEES

ID	Age	Years' Experience	No of projects	Job Title	Work Environment	Recently used Agile methodology	Team Size of the most recent project
1	Over 21	3-4 years	10+	UX Designer	Hybrid	A hybrid of Scrum and Kanban	1 - 3
2	Over 21	1-2 years	10+	UX Analyst	Hybrid	Scrum	4 - 6
3	Over 21	1-2 years	6 -10	Software Developer	Office	Scrum	4 - 6
4	Over 21	1-2 years	10+	Project Manager	Remote	Scrum	7 - 10
5	Over 21	4+ years	10+	Manager	Office	A hybrid of Scrum and Kanban	4 - 6
6	Over 21	4+ years	6 - 10	Technical Lead	Hybrid	Scrum	4 -6
7	Over 21	1-2 years	1 - 5	Software Developer	Hybrid	Scrum	4 - 6
8	Over 21	3-12 months	1 - 5	Tester	Office	Scrum	7 - 10
9	Over 21	1-2 years	1 - 5	Software Developer	Hybrid	Scrum	7 - 10
10	Over 21	4+ years	10+	Business Analyst	Remote	Scrum	10+
11	Over 21	4+ years	10+	UI Designer	Hybrid	Scrum	4 - 6
12	Over 21	4+ years	10+	Software Developer	Office	A hybrid of Scrum and Kanban	1 - 3
13	Over 21	4+ years	10+	UI Designer	Hybrid	Scrum	4 - 6
14	Over 21	1-2 years	6 - 10	Software Developer	Hybrid	Scrum	1 - 3
15	Over 21	3-4 years	6 - 10	UX Designer	Hybrid	Kanban	1 - 3

V. RESULTS

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 [14] and [72] 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. Additionally, the interviewees were asked to list the criteria they used to determine project success, and this resulted in an additional success criterion being identified. These findings were used to develop the framework shown in Fig. 2.

A. Agile Software Development Project Success Criteria

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 literature review 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. Furthermore, 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 [64].

B. Organizational Factors

The organizational factors category includes factors relating to the organizational structure, organizational environment and administrative climate of the company [7], [14], [25], [43], [44], [66]. The organizational factors identified in this study are managing customer expectations, communication, rapid feedback, organizational culture, facilitative

leadership, conducive, collaborative work environment, and societal culture. These factors are discussed below.

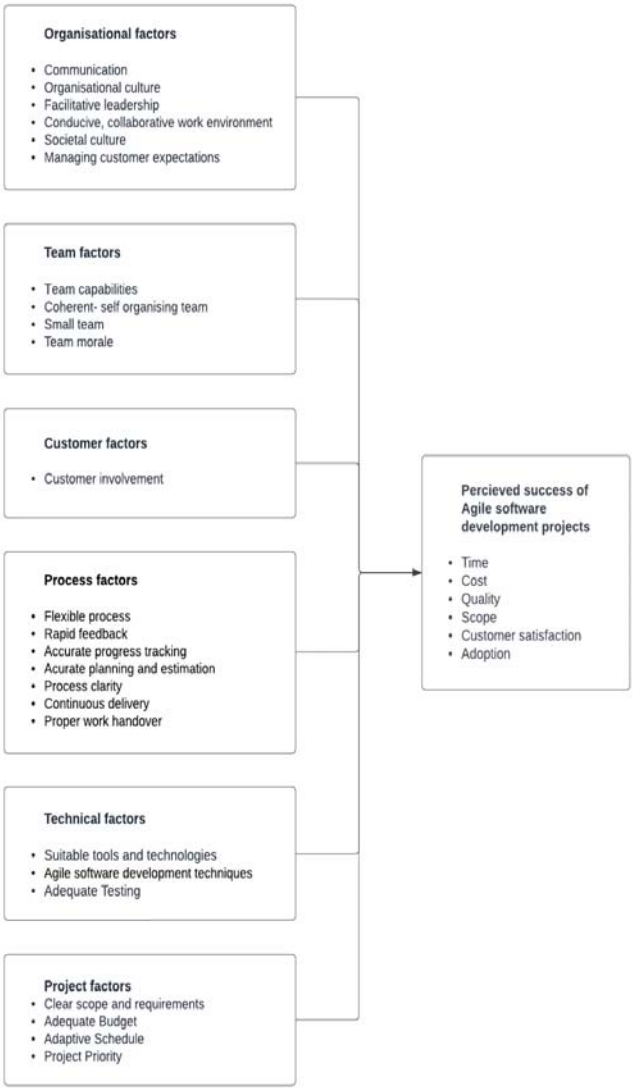

Fig. 2. Framework for the CSF of Agile software development in the South African software development industry.

Managing customer expectations.

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

Customer expectations significantly affect project success as this influence's 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 organization 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 will fail 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 organization 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).

The organization 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).

Communication

Communication was ranked seventh in the frequency-based ranking, and it is the third-highest-ranked organizational factor. One interviewee stated, "*Communication is one of the biggest factors within teams and the organization*" (14). A common thread in the findings is that the interviewees emphasized effective communication. 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 essential…having methods and techniques in place to facilitate clear and as close to real-time communication as possible… what determines success or failure often in communication is how comfortable it is for people to communicate in real-time.*" (5).

The case study organization 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 organization needs to ensure that there is a way for people to communicate effectively*" (4). What Interviewee 4 mentions may apply to all Agile organizations with collocated teams. A method to aid in effective communication, particularly online communication, that was brought up is using a specific tool throughout the organization for communication. Interviewee 8 states, "*Using ClickUP and Slack has been a massive help in organization and communication, so I see the need for them*" (8). The primary communication tool used in this organization is Slack. However, organizations can pick the tool that suits them best.

Organizational Culture

The organizational culture may be characterized as a collection of organizational elements or variables impacting a company's Agile software development [2]. According to the findings, organizational culture is the most influential critical success factor, as it was the highest-ranked. "*I would say that the organizational culture is very important. How the culture is within the organization, and if it benefits everyone*" (9).

Interviewee 5 highlighted 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 organizational culture would have the following attributes mentioned by the Agile practitioners: adaptability to change, knowledge sharing, a suitable organizational structure, and organizational buy-in into using Agile.

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

Another aspect of organizational culture is knowledge sharing. It is perceived that "*Organizations 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 organization, and as mentioned by Interviewee 2, "*This might relate again to organizational culture, having a knowledge-sharing culture. So, for instance, we have a knowledge share opportunity at an organization 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 organization.

The following attribute of organizational culture is the organizational structure. According to the findings, a flat organizational culture with less power distance suits Agile software development. "*I think organizational culture is a critical thing … I think it needs to be flat in terms of organizational structure because there is no strict chain of command. In an Agile environment, the best ideas must win … it is important that everyone's ideas matter and that no hierarchy goes against that.*" (5)

An organizational 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 organization, but that usually informs the structure of the project... So, if there is like much bureaucracy or if there is, like, a massive reporting hierarchy for one project, it could be quite difficult to work with*" (1).

Lastly, an organization needs to buy into the idea of using Agile, which goes hand in hand with ensuring the organization 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 organizations believe that Agile means no rules, and we can make the rules as we go. So, it is a good thing for organizations, management in organizations, 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)

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 organization 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).

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.*"

The literature in the literature review 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 organization who have a facilitative approach and exhibit the attributes discussed in this section may improve the chances of project success.

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 [79]. In the organization, some team members work from the office, others work from home, and others alternate.

Individuals work differently, and the organization 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 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 many 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 organization, 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 organizational 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*".

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 behaviors [75]. Suppose a project is being developed for a target audience with a different culture than the organization 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, cultures, and 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 literature review as a critical success factor, the study by [12] 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.

C. Team factors

The team category covers factors relating to the people who manage and execute the project [7], [14], [25], [43], [44], [66]. The factors include the team members' characteristics and properties necessary to complete the task [25]. The team category also examines practices relating to the conduct and working style of those participating in the software development process [6]. 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-organizing team, small team and team morale. These factors are discussed below.

Team Capabilities

Team capability refers to using knowledge and the conditions that enable teams to do their jobs successfully [75]. This factor was the second highest-ranked team factor in eighth place in the overall ranking. According to [14], 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. "*Often, we talk about needing to*

deliver this work in this amount of time, which is imposed on the team. However, 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 organization 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 valuable. "*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 can 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).

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)

Coherent - Self-Organizing team

The interviewees mentioned that the project team must be self-organizing and not reliant on management. In the Agile approach, a self-organizing team can coordinate their work independently and entirely control the development process [72]. The attributes of a coherent, self-organizing team identified in the findings were team cohesion and coordination.

Coordination in a self-organizing 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 organize 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-organizing, there needs to be cohesion. According to the interviewees, how well a team can work together dramatically impacts project success as it can affect other aspects such as team morale and productivity. "*The biggest thing that can impact project success is how people work together*" (10).

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).

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 organize 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"*.

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 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.

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 literature review 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, it was 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).

Other aspects of team morale mentioned were attitude, mindset, drive and motivation. 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)

The interviewees mentioned some ways individuals can be motivated. The first is through the compensation or rewards the organization 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 well, you do get a reward in terms of words of affirmation, which usually translates 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).

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 organization must develop projects with motivated people, providing the conditions and support they require and trusting them to do their work well [22]. 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.

D. Customer factors

Customer factors include factors relating to the people who sponsor the project or will use the product [7], [25], [66]. The customer factor identified in this category was customer involvement.

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 [14]: 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 made success is with hands-on clients, 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 organization or all three. Each role would need to be involved in the project to increase the likelihood of project success. *"The business sponsor also always needs to be reachable and available to give the input because that is another thing that can break a project 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, or a person that was not involved in the project's inception. Then, they might miss the mark and sign off on something Incorrect"* (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 emphasize 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 improving on anything. So 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.

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 organization. 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 existing literature. Having as much direct communication as possible between customers and development is vital because direct communication is efficient and incurs fewer losses [73].

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. The ranking emphasizes the importance of customer involvement in a project. Customer involvement is vital in Agile software development [2].

E. Process Factors

The process factors category covers factors relating to how project activities are carried out [7], [14], [25], [43], [44], [66]. The following process factors were identified: flexible process, accurate progress tracking, accurate planning and estimation, process clarity, continuous delivery, and proper work handover.

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 literature review 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"* [22:1].

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). Organizations 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 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 possible, would be beneficial. However, priorities are often very different for particular problems or scenarios where there may be a tighter deadline or a lower-risk project. So, you cannot have a one-size-fits-all process."* (5)

The project managers or team leaders 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 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 development organization, 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)

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: *"You develop software better with 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 organizational 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 organization 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.

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 organization 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 important, and our current one works well"* (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 organization can use a ticket system that is suitable for their organization.

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.

Reference [28] states 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, 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. Planning is 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 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 [22]

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 planning is vital, and understanding responsibilities, like where 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 [76]. 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, the goal of that specific task and 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. *"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 must try to use whatever means to ensure that the planning and related activities are done as accurately as possible.

Process Clarity

When a process to be followed in a project has been decided on and set in place, it is critical 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 a 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.

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 38]. According to [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 [59].

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).

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. Many miscommunications can happen in a project during handover that affects success. Interviewee 1 says, "*In each of the 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"*.

Thus, 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 going to 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.

F. Technical Factors

The technical factors category covers the tools, technologies, or techniques used in the project [7], [14], [25], [43], [44], [66]. It groups factors related to the engineering process of the software under development [6]. The technical factors identified in this study were suitable tools and technologies and adequate testing.

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 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 [8;44] .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)

Using advanced tools and technologies was identified as a critical success factor in the literature review. Conversely, the Agile practitioners in this study emphasized using suitable, familiar technologies over advanced technologies. [8] 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 [13] that found that using suitable technologies and tools positively affects Agile project success.

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 it 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 [53]. *"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.

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 [14] and [72], 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 [22]. 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 [77]. Interviewee 7 highlights the importance of having consistent coding standards: "*I think coding standards are fundamental. Working on a feature that needs to be integrated with another feature developed by someone else is very common. If the development process is the same, it will not matter because we will likely have a common coding standard used throughout the project"* (7). Consistent coding standards allow developers to integrate their work with the work of others easily, which can save the team time. Additionally, 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.

G. Project Factors

This category includes the factors relating to how project activities are carried out [7], [14], [25], [43], [44], [66]. It comprises factors related to the nature of the project during its development [6]. The project factors identified in this study were clear scope and requirement, adequate budget, adaptive schedule, and project priority.

Clear scope and requirements

A clear scope and requirements were 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 developers to be able to work on this one task they are given, they first need to be given some of that overview of what it is they are building and what the full system is"* (15).

Several interviewees emphasized 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? It is imperative 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 ensuring you understand what they need makes it much easier to stick to what you have planned"* (15).

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.

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, [14] 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 [14] stated, it might not be the most important CSF for organizations to focus on; however, it is still a critical success factor of Agile Software development.

Project Priority

According to the interviewees, each project has a different priority for the customer and the organization. *"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 organization, 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 organization 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 be neglected.

H. Hierarchy of Importance

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 [105].

The free association allows the researcher to process data produced directly from the free expression of individuals [105]. 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 [54], [48], and [100]. In their ranking, [100] 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 VI. 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.

The factor with the highest frequency and thus deemed the factor with the most significant influence on Agile project success is organizational culture. 10.3% of the mentions by interviewees were either on organizational culture or its attributes. Accordingly, it was also the highest-ranked organizational 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, organizational culture, was only 0.3%; thus, accurate planning and estimation and customer Involvement are also deemed significant influences 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 23rd, 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. 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 organizational categories. Thus, the team and organizational categories had the lowest ranking factors but the highest number of factors in the top ten. The organizational category had six identified factors, three of which were in the top ten: organizational 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-organizing team. These three factors were ranked fourth, eighth and ninth, respectively. Thus, the team factor had most of its factors in the top ten, but the organizational factors were in higher positions than the team factors. Therefore, it can be argued that the organizational 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 19th, respectively. Thus, the technical category does not have factors with the greatest nor the most negligible impact, similar to the process category.

I. Benchmarking the findings with the Literature Review Findings

At the beginning of the study, 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 organization. This section evaluates the similarities between the CSF identified in this study and the literature review. Fig. 3 depicts the overlap of critical success factors in this study versus those identified in the literature review in a Venn diagram.

Sixteen factors identified in this study were also identified in the literature review. The common factors in the organizational factor category were communication, organizational culture, conducive collaborative work environment, quick decision-making, and societal culture. In the team category, the common factors were team morale, coherent self-organizing 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 organizations. Of the overlapping factors, organizational culture, accurate planning and estimation, customer Involvement, team morale, communication, team capabilities and coherent – self-organizing teams are in the top ten. Thus, seven of the ten most highly ranked factors that the Agile practitioners mentioned are also considered critical

TABLE VI

FREQUENCY-BASED RANKING OF THE CRITICAL SUCCESS FACTORS

Rank	Critical Success Factor	Total number of mentions (n – 301) in the interviews		Category
		Combined Frequency	Percentage	
1	Organizational culture	31	10.3%	Organizational
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%	Organizational
6	Communication	16	5.3%	Organizational
7	Flexible process	16	5.3%	Process
8	Team capabilities	16	5.3%	Team
9	Coherent – self-organizing 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%	Organizational
14	Facilitative leadership	9	3.0%	Organizational
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%	Organizational

in other organizations 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 authors deemed the factors to be different since, according to the interviewees, an organization choosing advanced tools and technologies, or the latest technology might not always mean that the tools are suitable for the project.

In the research by [84], they suggested that strong executive support and project type do not influence the success of Agile projects. However, other researchers [7;13;22;53] concluded otherwise and deemed executive support a critical success factor. The findings in this study support [84] that strong executive support and project type were not identified in this research as Critical success factors. In this study and similar studies by [8] in Malaysia and [33] in Sri Lanka, and [12] in South Africa, the organization's work environment was identified as a Critical success factor. However, according to [84], the environment of an organization does not strongly influence the success of Agile software development projects in former Yugoslavia IT organizations. The findings being different in the various location suggests that the organization's location might affect the importance of the work environment. Based on this study's findings, the researchers recommend that South African software development organizations pay attention to the environment they foster for their Agile software development projects. They must foster a collaborative environment conducive to team member collaboration and productivity.

The study by [13] identified having a suitable progress-tracking mechanism as an attribute of a factor Project management process. However, this study identified it as a stand-alone factor and expanded to become accurate progress tracking. When an organization sets up a tracking mechanism, they must encourage employees to use it and update it frequently to have an accurate view of project progress.

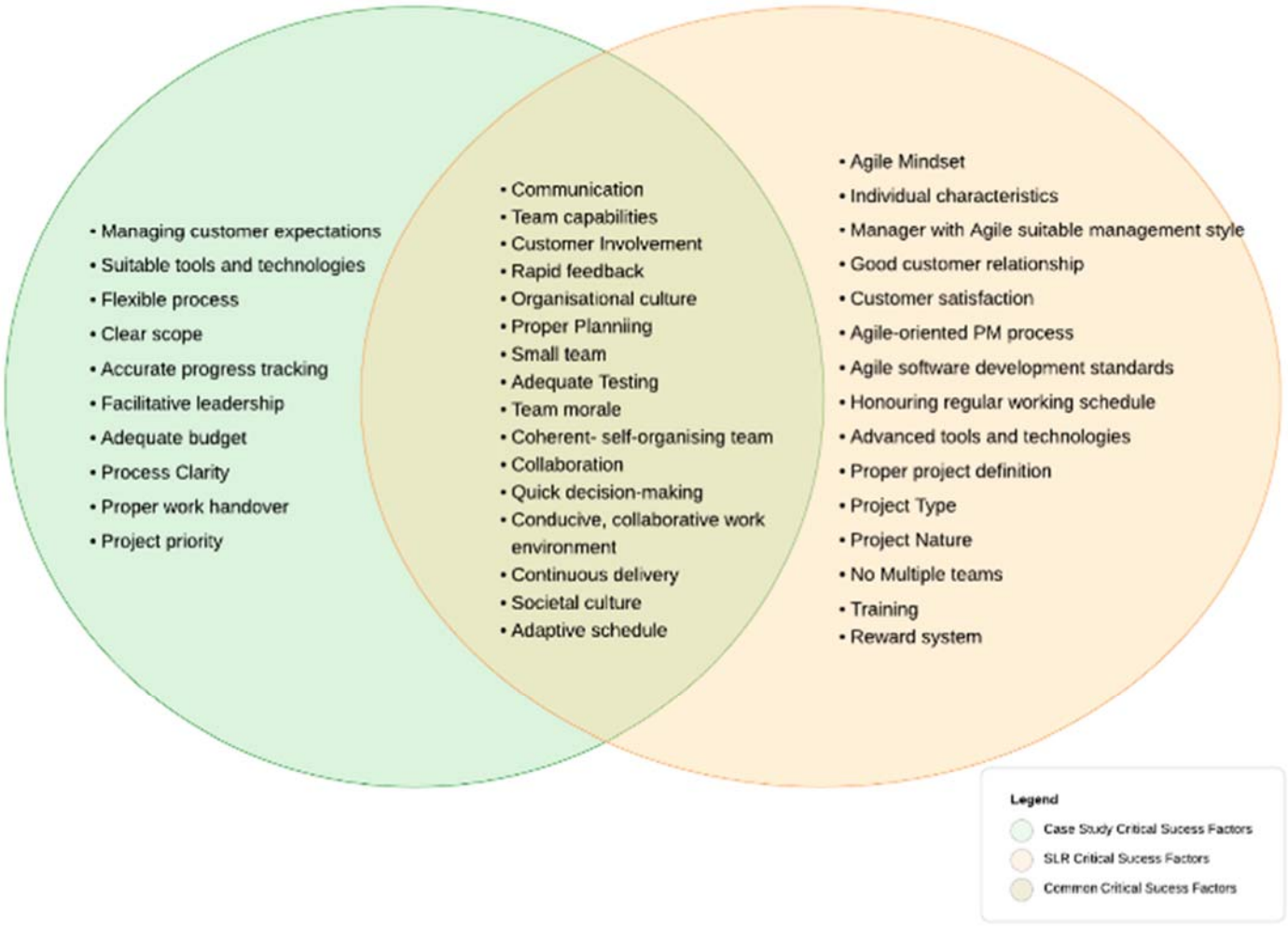

Fig. 3. Venn diagram showing the overlap of critical success factors identified in this study versus those identified in the literature review.

This study shows that Agile practitioners believe using suitable, familiar tools and technology is more beneficial to success than advanced tools and technology. Although, most research in the literature review mentioned using advanced technology [8]. The common factor addressed in their study was using advanced technology, and they concluded that balancing a good work environment with advanced technology will result in considerable improvements in success rates. However, our findings align with a similar but quantitative study conducted in South Africa [12], highlighting that Agile users' performance expectancy to practice Agile methods highly depends on using suitable technologies and tools, positively affecting Agile project success.

VI. Conclusion

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 Twenty-five critical success factors to the six categories identified: organizational, team, customer, process, technical and project.

The Identified critical success factors in the framework answered the Sub-research question 1. What are the critical factors in the success of Agile software development recognised by Agile practitioners in the South African software development industry?

The framework includes some new and expanded CSFs that were not identified in existing literature ad these are: managing customer expectations, facilitative leadership, flexible process, accurate progress tracking, process clarity, proper work handover and adequate budget. Since these factors are based on the perception of South African Agile practitioners it is recommended that Agile software development organizations make the following changes to address these new factors:

- Managing customer expectations by communicating to customers, sponsors, the users or the customer representatives the Agile methodology being used and what it entails in terms especially the frequency and format of deliverables.
- Having facilitative leaders that trust the team without much intervention, empowers the team to

make decisions, takes accountability for failures and successes and is open to suggestions from the team.
- Having a flexible process that is adaptive based on the project and making process adjustment based on team feedback.
- Accurately tracking progress by using a suitable ticketing system that allows the team to update progress accurate.
- Having a clear process that is easy to follow and well communicated to the team.
- Having proper work handover which includes handover meetings or hand over documentation.

Secondly a 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. This answered Sub-research question 2. What is the priority of the identified critical success factors?

The ranking showed that, according to the Agile practitioners in the case study, organizational culture is the CSF with the most significant influence on Agile project success. Thus, a recommendation is made to Agile software development organization to have an organizational culture that fosters success. This entails establishing a flat organizational structure where there is less power distance. Furthermore, cultivating a culture that promotes knowledge sharing, is adaptive to change having the whole organization buy-into Agile. Further insights from the ranking showed that the team and organizational category had the most factors in the top 10 ranking. Thus, it is recommended that Agile practitioners in South African software development focus on managing the identified organizational and team factors.

Thus, the research can be an eye-opener to other Agile practitioners, particularly managers and leaders in Agile software development organizations. Research in this area was lacking in the South African context, and it is believed that this addition to the scientific body of knowledge will provide information that if used can increase the probability of success of Agile software development projects in the South African software development industry.

A. Limitations

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 organization were involved, so casual inferences cannot be made from the data as alternative explanations cannot be ruled out. The case study organization was chosen because it had organizational members available and was willing to participate. Additionally, it was accessible to the researcher. A multi-case study can be done in future research.

B. 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 organization 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 organization where the employees work remotely or in one location. Additionally, it may be valuable for future research to investigate the interrelationships and interdependence between critical success factors success. This area of research will help develop a holistic view of Agile software development in the South African software development industry.

APPENDIX A - INTERVIEW STRUCTURE

Identification and prioritization 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 three months (Stop interview)
- Between 3 and 12 months
- 1 - 2 years
- 3 - 4 years
- Over four 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 ten 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: organizational factors, team factors, customer factors, process factors, technical factors, and project factors.

3.1. In your experience, what factors (factors relating to the organizational structure, organizational 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?

REFERENCES

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

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

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

[4] Aldahmash, A. M. 2018. A review on the critical success factors of Agile software development: an empirical study. University of Southampton.

[5] Alvarez, A. & Sánchez, B. B. 2022. 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.

[6] Arcos-Medina, G. & Mauricio, D. 2020. Identifying factors influencing Agile practices for software development. Journal of Information and Organizational Sciences, 44, 1-31.

[7] Asnawi, A. L. 2012. Investigating the adoption of and success factors for Agile software development in Malaysia. University of Southampton.

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

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

[10] 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.

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

[12] Chiyangwa, T. B. 2017. Modelling the critical success factors of Agile software development projects in South Africa.

[13] 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.

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

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

[16] 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.

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

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

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

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

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

[22] 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.

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

[24] Hummel, M. & Epp, A. 2015. Success Factors of Agile Information Systems Development: A Qualitative Study. 2015 48th Hawaii International Conference on System Sciences, 5-8 Jan. 2015 2015. 5045-5054.

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

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

[27] 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.

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

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

[30] Kelle, E. V., Visser, J., Plaat, A. & Wijst, P. V. D. 2015. 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.

[31] Enhance UX to account for distributed Connect architecture, M. & Werner, M. 2011. Interpretive case study research: Experiences and recommendations. MCIS 2011 Proceedings, 3-5.

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

[33] 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.

[34] Langkos, S. 2014. Chapter 3 - Research methodology: Data collection method and Research tools.

[35] Lee, A. S. 2004. Thinking about social theory and philosophy for information systems. Social theory and philosophy for information systems, 1, 26.

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

[37] Marnewick, C. 2012. A longitudinal analysis of ICT project success. Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference, 2012. 326-334.

[38] 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.

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

[40] Miler, J. & Gaida, P. 2019. 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.

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

[42] 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.

[43] Myers, M. D. 2019. Qualitative research in business and management, Sage.

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

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

[46] 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 an interview with school children. Revista brasileira de enfermagem, 71, 228-233.

[47] 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.

[48] 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.

[49] Noteboom, C., Ofori, M., Sutrave, K. & El-Gayar, O. 202. 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.

[50] Oates, B. J. 2006. Researching Information Systems and Computing London Sage Publications

[51] 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.
[52] Pereira, J., Varajão, J. & Takagi, N. 2022. Evaluation of information systems project success–Insights from practitioners. Information Systems Management, 39, 138-155.
[53] Peters, Y. D. 2020. Critical success factors influence the performance of Agile software development methodologies in organizations.
[54] 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).
[55] 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.
[56] Riaz, M. N., Mahboob, A. & Buriro, A. 2018. Social success factors affecting the implementation of Agile software development methodologies in the software industry of Pakistan: an empirical study. International Journal of Advanced Computer Science and Applications, 9, 94-98.
[57] Ries, E. 2009. Minimum viable product: a guide. Startup lessons learned 3.
[58] Rockart, J. F. & Crescenzi, A. D. 1984. Engaging top management in information technology. Sloan Management Review (pre-1986), 25, 3.
[59] 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.
[60] Rubin, H. J. & Rubin, I. S. 2011. Qualitative interviewing: The art of hearing data, sage.
[61] 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.
[62] 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.
[63] Saunders, M., Lewis, P. & Thornhill, A. 2007. Research methods. Business Students 4th edition Pearson Education Limited, England.
[64] Shakya, P. & Shakya, S. 2020. The critical success factor of Agile methodology in the software industry of Nepal. Journal of Information Technology, 2, 135-143.
[65] Shameem, M., Khan, A. A., Hasan, M. G. & Akbar, M. A. 2020. Analytic hierarchy process-based prioritization and taxonomy of success factors for scaling Agile methods in global software development. IET Software, 14, 389-401.
[66] Shameem, M., Kumar, C., Chandra, B. & Khan, A. A. 2017. 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.
[67] 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.
[68] 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.
[69] Shenhar, A. J., Dvir, D., Levy, O. & Maltz, A. C. 2001. Project success: a multidimensional strategic concept. Long range planning, 34, 699-725.
[70] 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.
[71] Stelzmann, E., Kreiner, C., Spork, G., Messnarz, R. & Koenig, F. 2010. 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.
[72] Syeda, S. H. 2018. An exploratory study to identify critical success factors of Agile systems engineering. The George Washington University.
[73] Tam, C., Da Costa Moura, E. J., Oliveira, T. & Varajão, J. 2020. The factors influencing the success of ongoing Agile software development projects. International Journal of Project Management, 38, 165-176.
[74] Thomas, J. 2008. 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.
[75] Timperi, O. P. 2004. An overview of quality assurance practices in Agile methodologies. T-76.650 Seminar in Software Engineering, 2004. Citeseer.
[76] Tona, C., Juárez-Ramírez, R., Jiménez, S., Durán, M. & Guerra-García, C. 2019. 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.
[77] Tyagi, S., Sibal, R. & Suri, B. 2022. Empirically developed framework for building trust in distributed Agile teams. Information and Software Technology, 145, 106828.

[78] Wagener, R. P. 2012. Investigating critical success factors in Agile systems development projects. North-West University, Potchefstroom Campus.
[79] Walsham, G. 1993. Interpreting information systems in organizations, Wiley Chichester.
[80] Walsham, G. 1995. Interpretive case studies in IS research: nature and method. European Journal of Information Systems, 4, 74-81.
[81] Wolf, C., Fu, Y.-C., Smith, T. & Joye, D. 2016. The SAGE handbook of survey methodology. The SAGE Handbook of Survey Methodology, 1-740.
[82] Yin, R. K. 1994. Discovering the future of the case study. Method in evaluation research. Evaluation practice, 15, 283-290.
[83] Wolf, C., Fu, Y.-C., Smith, T. & Joye, D. 2016. The SAGE handbook of survey methodology. The SAGE Handbook of Survey Methodology, 1-740.
[84] 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.
[85] Joseph, N. M. C. & Santana, M. 2016. Agile software development and IT project performance in South Africa: a positive relationship. The 25th International Conference for Management of Technology, 2016. 335-355.
[86] 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.
[87] Xiao, Y. & Watson, M. 2019. Guidance on conducting a systematic literature review. Journal of planning education and research, 39, 93-112.
[88] Webster, J. & Watson, R. T. 2002. Analyzing the past to prepare for the future: Writing a literature review. MIS quarterly, xiii-xxiii.
[89] Hesse-Biber, S. & Leavy, P. 2011. The Practice of Qualitative Research. The Practice of Qualitative Research. Thousand Oaks.
[90] Hussein, A. 2009. The use of triangulation in social sciences research. Journal of Comparative Social Work, 4, 106-117.
[91] Heale, R. & Forbes, D. 2013. Understanding triangulation in research. Evidence-based nursing, 16, 98-98.
[92] Etikan, I. & Bala, K. 2017. Sampling and sampling methods. Biometrics & Biostatistics International Journal, 5, 00149.
[93] Ishak, N. M. & Abu Bakar, A. Y. 2014. Developing Sampling Frame for Case Study: Challenges and Conditions. World journal of education, 4, 29-35.
[94] Boddy, C. R. 2016. Sample size for qualitative research. Qualitative Market Research: An International Journal.
[95] 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.
[96] Bogopa, M. E. & Marnewick, C. 2022. Critical success factors in software development projects. South African Computer Journal, 34, 1-34.
[97] 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.
[98] 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.
[99] 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.
[100] 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.
[101] G. Lee and W. Xia, “Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data on Software Development Agility,” MIS Q., vol. 34, no. 1, pp. 87–114, 2010.
[102] Reunamäki, R. and Fey, C.F., 2023. Remote Agile: Problems, solutions, and pitfalls to avoid. Business Horizons, 66(4), pp.505-516.
[103] Tetteh, S.G., 2024. Empirical Study of Agile Software Development Methodologies: A Comparative Analysis. Asian Journal of Research in Computer Science, 17(5), pp.30-42.
[104] Rockart, J. F. & Crescenzi, A. D. 1984. Engaging top management in information technology. Sloan Management Review (pre-1986), 25, 3.
[105] 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.

A. Gwangwadza is a master’s in information systems graduate from the University of Pretoria, South Africa. Her research interests lie in Agile software development and project management.

R. Hanslo is a Lecturer at the Academy of Computer Science and Software Engineering, University of Johannesburg, South Africa. He has more than 20 years of industry experience and have been an Agile practitioner for more than a decade. His research interests lie in software engineering and artificial intelligence.