UNIVERSITY OF KWAZULU-NATAL

Software development project success: Perspectives of project managers and developers in a South African Bank

by

Aruna Seerpath

213503484

A dissertation submitted in fulfilment of the requirements for the degree of
Master of Commerce

School of Management, IT and Governance
College of Law and Management Studies

Supervisor: Miss C. Sue Price

2020

Declaration

I, **Aruna Seerpath**, declare that

i. The research reported in this dissertation, except where otherwise indicated, is my original research.

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

iii. This dissertation does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons.

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

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

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

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

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

Signed:  Date: 23 April 2021

Acknowledgements

Firstly, I would like to express my deepest gratitude to my supervisor Miss Sue Price who has guided me throughout my post graduate studies.

Secondly, I would like to thank Kiran for motivating me, making me laugh and uplifting me throughout the course of my masters.

Thirdly, this work would not have been possible without the love and support from my mum and sister. Their words of encouragement have always provided me with a sense of determination to do my best and make them proud.

Lastly, my sincere thanks to my grandparents, whose work ethic has always inspired me to do great things in life.

I would also like to thank the examiners whose suggestions have improved this work.

Abstract

Many researchers have attempted to define software development project success. While common factors have been found, the success of software projects in all industries cannot be judged by the same factors. Success is complex and means different things to different people. This study investigated what factors project managers and software developers, in a South African Bank, believed contributed to software development project success. The reasons why software developers and project managers were investigated were because software developers have a technical background whilst project managers have a business background. There were 41 respondents consisting of six project managers and 35 software developers that answered the questionnaire. Six participants were interviewed: four developers and two project managers. A case study research was adopted for this study. The results showed that there are many significant factors that contribute to success and failure.

The study was divided into four categories to investigate the different success factors. These categories were individual, team, organisational and project. The most critical success factor from this study was the projects being strictly according to the on brief. This showed the software being built must fulfil the requirements that were defined in the project brief. The least important success factor was employees having more than ten years of experience. This showed that this group of respondents do not believe that such a high level of experience is directly related to project success. Respondents were also asked what they thought would lead to software project failure. The factor that was considered most significant was not understanding requirements.

An individual factor that stood out was the level of skill of the staff. Both groups of employees believe that having the appropriate skills to do one's job, contributes significantly to the success of a project. The only factors where project managers and developers responded significantly differently were when they were asked about work life balance and job satisfaction. Developers seemed to consider these factors more important than project managers. Project managers and software developers considered the scope and size of a project critical to project success as it ranked as the second most critical factor. Other critical success factors that appeared in the top five were the level of skill of staff, clearly defined business objectives and understanding requirements.

This study contributes to existing literature as it shows that success factors that other researchers found cannot be generalized to all individuals, teams, organisations and projects. This study investigated what factors, project managers and software developers in this digital banking business unit considered to contribute to the success of a project. It concluded that there are many factors that contribute to success however, projects being on brief was the most important factor. Even though this study only investigated the views of project managers and software developers, there are other IT professionals that could offer different insights as to what success means to them. This study could therefore be expanded to include business analysts, system analysts and testers.

Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Introduction

Software can be described as a collection of instructions, data and programs that tell a computer system how to work. The process of creating software includes the documentation of software requirements; programming the solution; testing; rectifying of any errors in code and maintenance of the software to produce an effective and efficient software solution (Ruparelia, 2010). The organised manner in which these steps are followed is essential during a software development project (Jurison, 1999).

Software development projects are shaped by a definite set of objectives and requirements. These objectives and requirements guide the project during the software development process. A measurement of how well these objectives and requirements are met provides an indication of whether a project is successful or not (Baccarini, 1999).

Unfortunately, not all projects are successful. In 1994, the Standish Group showed in the Chaos Report that 16% of all software projects were successful. Twenty years later, the Standish Group reported that the average rate of successful projects is 29%, 52 % of the projects are challenged and 19% fail outright (Hastie & Wojewoda, 2015). In 2016, Hastie & Wojewoda, (2015) stated that 31.1% of projects would be cancelled before they have been completed.

While the percentage of failed projects has decreased since 1994, far too many software development projects are still challenged or failing. The common reasons are not catering for changing requirements in the project plan; not anticipating the fact that more resources would be needed; not allocating enough time for detailed requirement analysis; and not allocating enough time to testing and reworking (Attarzadeh & Ow, 2008). With such high levels of failure, it is important to prevent failure and understand what makes a software project successful.

Some of the factors affecting project success include whether or not projects were on time, on budget and exhibited good quality and whether users were satisfied with the end product. This is

usually referred to as the iron triangle (Hastie & Wojewoda, 2015). In addition, the extent to which users, management, team members and project managers are involved during the software development project process affects project success (Tarawneh, 2011). The extent to which requirements are well defined and understood by all members in a software development team is another important factor. Based on these and many other factors, it can then be determined if a project was successful.

What can appear successful to a customer or stakeholder may not necessarily be true for the project team that was involved in the software development process (Verner & Evanco, 2003). Customers may be happy with the end product as it has been developed based on their requirements. Stakeholders may also be content that the product was released to the customer (Beck et al., 2001). The project team, on the other hand, may or not may be fully satisfied with the process that occurred during development.

The project team consists of project managers, software developers, software testers and business analysts. Some of the people in the team may work longer hours and on week-ends to complete their part of the project. This may result in poor work-life balance, as well as team members working overtime. The need to work overtime may be as a result of issues such as changing requirements or requirements which were not specified correctly. It could also be because the different members in their team did not produce good quality work; they lacked the necessary skills to complete tasks; or the project may have not been managed correctly (Verner & Evanco, 2003). As a result, team members may feel dissatisfied with their jobs, which creates a negative working environment and could lead to further project failure. This means that different members in a project team may have different opinions on what constitutes success.

The success of a software project can also be affected by the software methodology adopted. For years, many traditional and agile approaches have been debated as to which approach is more appropriate for software development. Agile is highly dependent on people and interactions thus has been found to be successful for smaller teams and projects (van Kelle, Visser, Plaat, & van der Wijst, 2015). Larger organisations and projects may have difficulty adopting agile approaches, therefore they prefer a traditional approach as there is less social intervention (van Kelle, Visser, Plaat, & van der Wijst, 2015).

Different IT professionals, even in the same team, may have different views as to what success means to them. This study investigated the views of project managers and software developers only. Project managers are more business oriented as they believe that if a project is on time, within budget and within scope, it is successful (Atkinson, 1999). Software developers are more technical and believe that if they can be creative in coming up with a solution for the project, then a project will be more successful (Linberg, 1999). The conflicting opinions are based on the fact that the project managers' main focus is on meeting the business objectives, whereas software developers are more guided by creativity and job satisfaction. These two groups of employees were therefore chosen as they represent views from a business perspective and from a technical perspective.

Many researchers have attempted to define software development project success. While common factors have been found, the success of all projects cannot be judged by the same factors. This may be influenced by different industries, such as finance, agriculture or educational; with different companies, people and situations achieving different degrees of success. The factors that commonly affect software development project success can be grouped by individual factors, team factors, organisational and project factors. The individual category includes the extent to which staff is skilled; employee experience; work-life balance and job satisfaction. In the team category, team dynamics such as communication, accountability, equality and motivation were investigated. The organisational category includes issues such as the extent to which management and stakeholders support the project; the extent to which business objectives are well defined; and the organisational culture of the company. Lastly, the project category looks at the extent to which requirements are understood; how changes to requirements are managed; software quality control; the availability of subject matter experts; the scope and sizing of projects; the project management iron triangle; how risks are managed; the methodology adopted and user involvement. These factors will be investigated for this study.

Software products have been developed in the banking industry to assist customers and staff with the tools that they need to improve banking. The banking industry strives to develop successful products and as a result new platforms for banking have emerged over the past few years. Internet banking has become an easier way of banking as it allows customers to satisfy their daily financial needs anywhere, at any time (Püschel, Mazzon, & Hernandez, 2010). As the rapid evolution in technology continues, newer and more efficient ways of banking emerge. This results in banks having to produce a number of software products to keep up with the ever-changing needs of customers.

1.2 Identification of problems/issues

Project managers and software developers have different opinions of what constitutes software development project success (Verner & Evanco, 2003). These two groups of individuals offer a different range of skills and play different roles in the team. Whilst developers are more technical, project managers are more business oriented hence these groups will be compared. The factors affecting project managers' and software developers' perceptions of project success within the South African banking sector are unknown.

1.3 Background

A South African bank has been identified as this research's project site as the banking sector is one of the major sectors in the country's economy. The South African bank is listed on the JSE and Namibian stock exchange. It is one of the largest financial institutions in South Africa. It provides banking, investment and insurance products and services to retail, corporate, commercial and public sector customers. The bank has digital platforms such as an online banking website, a cellphone banking platform and a mobile banking application. The digital banking business unit follows a strict software development process for each of its projects, with the aim of adding new features to their digital banking platforms. One of the units that falls under the digital banking business unit is the mobile applications unit. The mobile application business unit is responsible for implementing efficient ways of banking on a digital platform. This mobile application business unit will be investigated for this study.

There are ten software development releases in a year, one per month from February to November and each release is made up of six to ten projects. When a project is initiated in the mobile application banking unit, it starts off with 12 weeks of business analysis, followed by four weeks of technical analysis, six weeks of development and six weeks of testing. The entire project team consists of approximately eight people: a project manager, a solutions architect; a systems architect; a business analyst; a systems analyst; a developer team leader; a developer; a test analyst team leader and a test analyst. This entire process is managed by project managers to ensure all deadlines and milestones are met. At any given time, a developer is working on one or two projects and project managers may be working on up to three projects in a month.

Business and systems analysts are assigned by the channel managers. They choose which business analyst is most suited for the project based on their level of experience, their familiarity with the new feature or their work load. In this banking unit, there are four software development teams. Each software development team is assigned to each of the releases, i.e. Team A works on February release, Team B works on March release, Team C works on April release, Team D works on May release and then Team A works on June release etc. The development teams work on releases on a rotational basis. The developer team leader of the assigned team will choose which of his/her team members should be assigned to the project. They are chosen in a similar way to how the business and systems analysts are chosen.

When a new project is logged with the mobile application business unit, either from the CEO from the business unit or from another business unit, the project first goes to scheduling to determine an appropriate date to implement the new project. Once a date has been set, the project manager starts the project and liaises with the analysts, architects, developers and testers. The analysis team consists of one business analyst and one systems analyst. The scope and sizing of projects is estimated by the solutions architects. The solutions and software architects, also known as the subject matter experts, usually have a meeting to discuss how the new feature fits in with, and adheres to the current architectural standards of the mobile application. After this meeting, the business analysts and systems analysts start to examine the requirements further. During the analysis phase, many meetings are held to understand how the new features will work and look. These meetings are guided by the solutions and systems architects to ensure that the new features add business value and improves customers' banking experience.

Once analysis is complete, development begins. During this time, weekly update sessions are held to assess the progress of the project. During the weekly sessions, developers may challenge what was advised by the solutions/ systems architects as well as what the initial requirement was. If changes need to be made to improve the project, it is usually updated on the BA's documentation before developers can make a change in code. After the six weeks of development, the project team starts integration testing. Integration testing is when the test analysts evaluate the new software features and ensure that the software produced is of good quality. Thereafter the project goes into quality assurance testing for two weeks before it is released to customers.

This South African bank's digital business unit is split into three sections, online, app and cell phone banking (dial string). The online section was given the best Internet banking award in 2017 as voted by customers and other companies. The bank was also voted the 'coolest bank in South Africa' in 2019 by consumers. To customers and other companies, the bank's software development projects are successful as they produce new and exciting features that make banking easier. However, no insight is provided as to whether the software development process is successful from the employees' points of view. Team members may or may not be content with the process, as well as the outcome. An investigation is therefore necessary to determine what factors are critical to the success of this bank's software development projects from a project manager's and software developer's perspective.

1.4 Research questions

1. How critical are individual factors for software development project success for project managers and software developers at a South African bank?
2. How critical are team factors for software development project success for project managers and software developers at a South African bank?
3. How critical are organisational factors for software development project success for project managers and software developers at a South African bank?
4. How critical are project factors for software development project success for project managers and software developers at a South African bank?
5. How can software development project failure be prevented, from a project manager's and software developer's perspective, within a South African bank?

1.5 Research objectives

- To determine the extent to which individual factors are critical for software development project success from a project manager's and software developer's perspective within a South African bank.
- To determine the extent to which team factors are critical for software development project success from a project manager's and software developer's perspective within a South African bank.

- To determine the extent to which organisational factors are critical for software development project success from a project manager's and software developer's perspective within a South African bank.
- To determine the extent to which project factors are critical for software development project success from a project manager's and software developer's perspective within a South African bank.
- To determine what factors can prevent software development project failure from a project manager's and software developer's perspective within a South African bank.

1.6 Conclusion

The aim of this study is to evaluate what success means to project managers and software developers within a South African bank. This chapter outlined the area of research, the problem to be investigated, the background for the study, the research questions and objectives. The next chapter provides a literature review that discusses various factors affecting software development project success.

Chapter 2: Literature Review

2.1 Introduction

The previous chapter provided an understanding of the research problem, the area of research and background, as well as the research questions and objectives. This chapter starts of by discussing what software development is and the methodologies that can be used to develop software. Project management is also defined and the project management techniques that are generally used. Thereafter it explores software development project success from project managers' and developers' perspectives as well as software development project success in financial institutions. Software development project failure is also discussed from the points of view of project managers and developers. Failure prevention techniques are also presented based on recommendations from previous literature. A conceptual framework for the study is then presented based on the findings from the literature discussed.

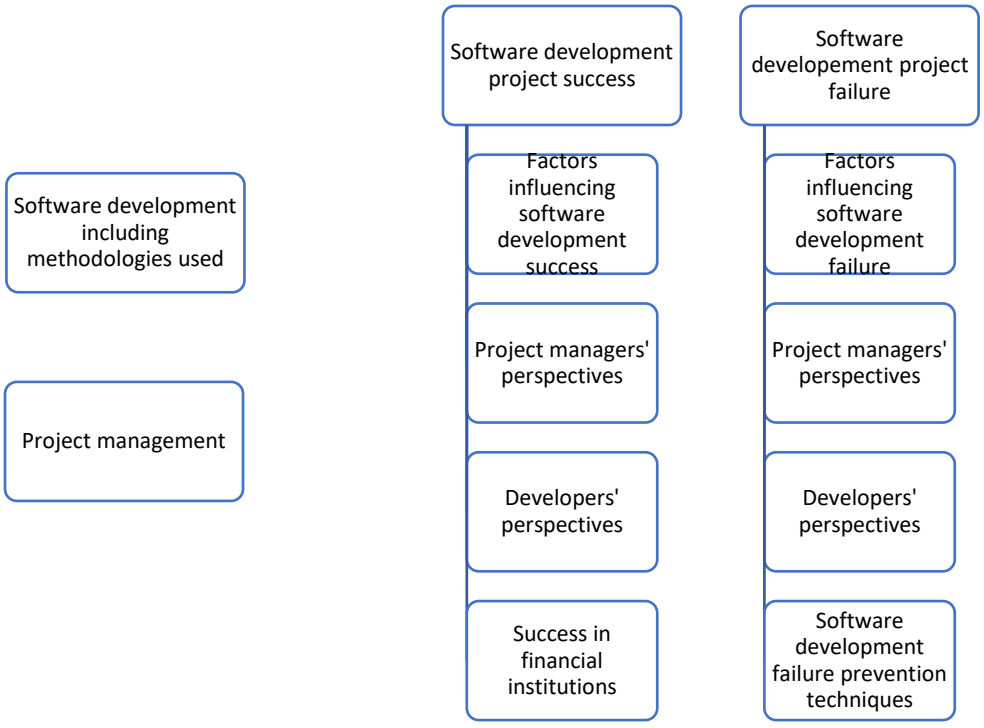

Figure 1: Literature review sections

2.2 Software development

Software development processes refer to a series of steps or phases that are followed to create software. These phases include requirements gathering, analysis, design, implementation, testing and maintenance. Requirements gathering involves sitting with the users, managers and stakeholders to understand what features they require in the new software (Bassil, 2012). Analysis of these requirements then follows, where the business and systems analysts dissect the requirements to understand how the new system will work. The new system needs to adhere to the company's architecture, as well as what would be the best solution for the business (Bassil, 2012). Thereafter, the design phase commences where software developers, software architects, usability specialists and analysts come up with a solution (Bassil, 2012). Implementation involves the developing or coding of the software. Testing of this software is a vital phase as software defects will be picked up and rectified and the quality of the software will be ensured (Bassil, 2012). Once the software has been implemented and tested, the new system is made available to users. At this point, users can begin using the new product. The last phase involves maintenance of this software, where updates may be made as well as enhancements to features. These phases essentially guide the development of a project. However, different software development models are chosen, based on different projects.

Software development models are chosen to assist with the development of a project according to its objectives. The various models have specific features that make them suitable for different projects. A model will be chosen depending on whether all the project requirements are known upfront, or whether the requirements are constantly changing; what the scope and size of the project is; what resources are needed; who the users of the system are; the extent to which customer feedback is required during development and when it is required; whether the architecture is fully understood or not; to name but a few (Stoica, Mircea & Ghilic-Micu, 2013).

2.3 Software development methodologies

There are many software development methodologies available. They can be divided into two categories: traditional and agile. Traditional methodologies are plan-driven, which involves the documentation of a complete set of requirements, high level architecture and full design development (Awad, 2005). Agile projects can be described as having the ability to adapt to change. Software models that allow for the changing of the project requirements during their

execution can be referred to as agile methodologies (Awad, 2005). These two methodologies are described below.

2.3.1 Traditional software development

Traditional software methodologies are guided by a few defining factors. Traditional methodologies focus on software correctness (Balsamo, Di Marco, Inverardi & Simeoni, 2004). A structured approach between phases is emphasised, where one phase, consisting of a definitive set of goals and tasks, must be complete before commencing on the next phase, producing functional software at the end of all the phases (Awad, 2005). Managers prefer traditional methods as they have more control over the projects. Traditional models consist of identifying, modelling, communicating and documenting all requirements and features of a system before any design and implementation work is started. This approach works well if requirements are not changing and the technology and architecture is well understood by developers (Paetsch, Eberlein & Maurer, 2003). Reworking software using a traditional method is expensive, as more time and resources will be needed. All requirements need to be defined upfront (Paetsch et al., 2003). This structured approach allows for the project to be measured easily and for deadlines to be met (Stoica et al., 2013). It also allows software to be more complete and to be designed more carefully, so it is suitable for critical systems such as medical equipment, nuclear reactors and amusement park rides.

There are, however, a few drawbacks in using this traditional approach. The structured nature of traditional models limits the inclusion of any changes requested by users during the course of the project, after the requirements engineering phase is complete. Requirements could possibly change if clients were unsure about what they needed in the system. Requirements may also change if the analysts or software development team did not understand the requirements thoroughly to begin with, or if the technology changes while the project is being implemented. Traditional methods appear to be more costly for smaller projects and teams (Munassar & Govardhan, 2010). They tend to take longer to complete as the process is structured and every phase must be complete and well documented before starting the next (Munassar & Govardhan, 2010). Some examples of traditional software development methodologies include the Waterfall model and V-Model.

2.3.2 Agile software development

Agile development is an iterative approach that emphasises rapid application development to produce functional software components after every iteration (Sharma, Sarkar & Gupta, 2012). Agile methods rely less on documentation and more on coding and are more easily adapted to handle frequent changes during the software development lifecycle (Paetsch et al., 2003). The changing requirements from users cause software development teams to change their strategies and approaches when developing software. This enables the team to cater for these dynamic business environments (Nerur, Mahapatra & Mangalaraj, 2005).

Agile software development is guided by the agile manifesto (Beck et al., 2001). The agile manifesto came about when software practitioners who worked with the different software development methodologies that catered for changing requirements attended a forum to discuss the term 'agile' (Beck et al., 2001). In this forum, different ideas were presented and argued. An agreement was then made on the rules to follow when developing software in an agile manner (Beck et al., 2001). After this forum, it was decided that agile methods should place more emphasis on individuals and interactions, working software, customer collaboration and responding to change rather than processes and tools, comprehensive documentation, contract negotiation and following a plan (Beck et al., 2001). The main advantage of adopting an agile approach over a traditional one is that it allows for one to develop products whose requirements change during development. The agile approach also responds well if the business environment changes, which in turn may also affect requirements.

Satisfying customer needs holds the highest priority in the agile manifesto; thus, customer feedback and collaboration is a major part of any development process (Beck et al., 2001). By allowing user experts to be readily available to the team, or even part of the team, allows them to test functionality incrementally (Cockburn & Highsmith, 2001). The agile manifesto emphasises that individuals who are part of the project team need to be motivated and encouraged. This is achieved by giving them the environment and support that they require and placing trust in them to get the job done. Thereafter, at regular intervals, the team reflects on how to improve on current processes and adjusts them accordingly (Beck et al., 2001). Working software is another factor, as it is the primary measure of progress (Beck et al., 2001). The agile manifesto states that attention to technical detail is imperative as technical excellence and good design enhance agility (Beck et al., 2001). Changing requirements are welcomed at any stage of development as the

agile process harnesses change, thus allowing the stakeholders to have a competitive advantage (Beck et al., 2001).

Lack of communication between team members, as well as between stakeholders and the targeted users, can lead to misunderstanding the requirements. In agile development, teams communicate better if information is passed to fewer team members. This reduces the time between making a decision and analysing the consequences of that decision (Cockburn & Highsmith, 2001). Therefore, by placing development teams closer together, replacing documentation with whiteboards and talking in person, communication is improved (Beck et al., 2001).

There are a number of software development methodologies that accommodate changing demands from the users. Some of the approaches in the agile methodology include extreme programming (XP), scrum and feature driven development (Leau et al., 2012).

2.3.3 Adoption of methodologies

A study was conducted that investigated which approaches are used in practice and which are combined (Theocharis, Kuhrmann, Münch & Diebold, 2015). Through the systematic literature review it was found that the scrum approach was adopted by multiple organisations. Scrum was also found to have been mixed with other agile approaches. It can also be seen that a reluctance existed to buy into agile completely, thus a hybrid approach was adopted that combined efforts of agile and traditional (Theocharis et al., 2015). Furthermore, they found that managers preferred a structured, plan oriented approach whereas developers preferred the freedom that agile offered. This showed that the hybrid approach offered benefits from both approaches and can be altered for a specific company or organisation (Theocharis et al., 2015).

Over the years agile methods seemed to offer more flexibility than traditional thus becoming more widely used. Whilst there is a stigma that agile is more suited for smaller projects and teams, van Kelle et al., (2015), found that the size of projects did not play a contributing factor in an agile project. Their study exposed that project success or failure is influenced by social factors.

Ahimbisibwe, Daellenbach & Cavana, (2017) conducted a study that used the methodology adopted as a moderating variable to compare how critical success factors contributed to project success. They found that critical success factors behave differently and have different relationships to traditional and agile methods Ahimbisibwe et al., (2017). This study showed that a methodology must be carefully chosen as it can drastically affect the outcome of a project. An organisation must choose a methodology based on their critical success factors as it guides what needs to be achieved in a project (Ahimbisibwe et al., 2017).

Khoza & Marnewick, (2020) also conducted a study to investigate the adoption of traditional and agile methods in South Africa. They found that in South Africa, traditional methodologies were used for larger projects. Waterfall projects in this sample of 617 projects were larger and longer than Agile counterparts. Waterfall spent 10.16% less than budget, but Agile spent 42.62% more than budget. Both spent more time than planned, but Waterfall spent less time (13.65%) than Agile projects did (22.4%).

2.4 Project management

A project can be thought of as the achievement of an objective which involves a series of phases, deadlines and milestones. Project management, by contrast, is the process of controlling the process to achieve the objective (Munns & Bjeirmi, 1996). Software project management can be thought of as the skilful integration of technology, economics and human resources (Boehm & Ross, 1989). Project managers have the difficult task of satisfying users, customers, the development team, the maintenance team and the stakeholders, all of which consist of very different people from contrasting backgrounds (Boehm & Ross, 1989). Their main aim is to find a way to manage strategic and tactical conflicts which can consist of setting goals, establishing milestones and responsibilities, as well as prioritising assignments, adapting to change and managing day-to-day issues that may arise (Boehm & Ross, 1989).

Project managers are responsible for guiding team members and managing projects to ensure that all goals are met and the project is a success. Project managers require both hard and soft skills. Hard skills can be thought of as technical skills where the project manager requires extensive knowledge of processes, tools and techniques to use when managing a project (Sukhoo, Barnard, Eloff, Van der Poll & Motah, 2005). Soft skills refer to how managers manage team members,

interact with customers and ensure employee and customer satisfaction. Soft skills also involve creating an environment that ensures that team members deliver quality products in the specified time frames, whilst meeting all of the stakeholder requirements (Sukhoo et al., 2005). Project managers must have extensive knowledge of the methodology that a team chooses to use. This translates into whether the project will be managed in an agile or traditional manner. This essentially guides how the project will be managed and sets out a plan for the project manager.

Various factors need to be taken into account during the management of a software project. These may include project planning, project cost estimation and milestone and goal tracking. These are discussed below.

2.5 Project management techniques

2.5.1 Project planning

Project managers begin the project management process by developing a plan of action. This includes creating work breakdown structures, timelines, staff assignments and creating a critical path to follow with regards to the activities. Analysis is also required to determine how much time will be needed for requirements gathering and analysis as well as accommodating any other risks (Attarzadeh & Ow, 2008). The common reasons that projects may fail are that the project plan did not cater for changing requirement; not anticipating the fact that more resources were needed; setting unachievable milestones and deadlines; and not allocating enough time to testing and reworking (Attarzadeh & Ow, 2008). Project planning may not guarantee the success of a project; however, the lack of it may end in project failure. Planning a project therefore reduces uncertainty (Dvir, Raz & Shenhar, 2003).

2.5.2 Project cost estimation

Software cost estimation models are used to create a budget; conduct trade-offs and risk analyses; plan and control projects; and conduct improvement analysis (Boehm, Abts & Chulani, 2000). Costs can be estimated by a number of different software tools. Financial experts, however, may prefer to conduct this analysis on their own (Attarzadeh & Ow, 2008). Cost estimation involves several steps which include sizing of major deliverables; comparing past projects that are similar to the current one in terms of size and scope; liaising with project management specialists; and

estimating how the project will handle changes (Attarzadeh & Ow, 2008). Sizing of a project includes understanding how many new features are required, what impact it has to the different teams and how much work is required to complete the project. Project sizing is usually done by the development team, and depends on what sizing models different companies use. Project managers use this as a guide to size deliverables (Attarzadeh & Ow, 2008).

2.5.3 Milestone tracking

Milestones can be thought of as a significant stage or event. In project management it describes what needs to get done, rather than explaining how to do it. Milestones therefore promote result-oriented thinking and not activity-oriented thinking (Dvir et al., 2003). Milestones allow for people to work towards something. They create a sense of completion and reaffirm the fact that the project is one step closer to being finished. Project managers are responsible for creating milestones, monitoring the progress towards reaching milestones and reporting truthfully on the success or failure to complete the milestone (Attarzadeh & Ow, 2008). Some milestones may include reviews of the project plan, requirements analysis, cost estimates, database designs, screen designs, system flows, development plans, testing plans, code inspections and many other deliverables that may be project specific (Attarzadeh & Ow, 2008). In traditional models, there are definite milestones at the end of each phase; in agile models, milestones are not clearly defined but can be thought of as the completion of a specific task (Attarzadeh & Ow, 2008).

Different project management techniques are adopted to suit the complexity of a project. Project management techniques are also dependent on the industry for which software is being developed, as well as the people that make up the software development team. The above-mentioned project management phases provide an overview on how software projects should be managed. They can be varied, and some phases may not be conducted thoroughly, due to the different techniques teams may use. Agile models, for example, do not give priority to documentation, so this aspect may not be fully completed. How a project is managed and implemented can affect whether the project is successful or not. A discussion follows on what failure and success of a software development project means.

2.6 Software development project success

Success is complex and means different things to different people and in different contexts. Whilst project success in general can be thought of as how well the objectives are met, factors leading to software development project success have not been agreed upon (Mäntylä, Jørgensen, Ralph & Erdogmus, 2017). In order to define a success measurement model, common factors can be found that are applicable to the project, team or organisation and a base model can be created (Mäntylä et al., 2017). Many researchers have attempted to create models to measure project success. The table below shows critical success factors as proposed by multiple researchers. Factors that appeared in multiple research papers are listed and a discussion on each of them follows below.

Table 1: Critical success factors found in the literature.

Critical Success Factor	References
Level of staff skills	Hastie & Wojewoda (2015); Octavianus and Mursanto (2018); Alqahtani et al. (2014); Fui-Hoon, Lee-Shang & Kuang (2001); Tarawneh (2011); Keil, Lee & Deng (2013); Mtsweni, Hörne & van der Poll (2016)
Employee experience	Dokko, Wilk & Rothbard (2009); Tarawneh (2011); Alqahtani et al. (2014); Hastie & Wojewoda (2015); Peter Norvig of Google (2014); Rehman (2006)
Work-life balance	Rehman (2006); Arif & Farooqi (2014); Heeks et al. (2017)
Job satisfaction	Halkos & Bousinakis (2010); Korrapati & Eedara (2010)
Communication	Ahimbisibwe et al. (2017); Estler, Nordio, Furia, Meyer & Schneider (2014); Sheffield & Lemétayer (2013)
Accountability	Reed & Knight (2010); McHugh, Conboy and Lang (2011)
Equality	Beecham (2014); Kivelä (2019)
Motivation	Takpuie & Tanner (2016); Daim et al. (2012); Estler et al. (2014)

Support from management and stakeholders	Fui-Hoon et al. (2001); Tarawneh (2011); Rezvani, Chang, Wiewiora, Ashkanasy, Jordan & Zolin (2016) Hastie & Wojewoda (2015); Octavianus & Mursanto (2018); Dezdar and Ainin (2011)
Organisational culture	Ahimbisibwe et al. (2017); Eldrandaly et al. (2015); Tarawneh (2011); Gu, Hoffman, Cao & Schniederjans (2014); Lee et al. (2016).
Clearly defined business objectives	Tarawneh (2011); Hastie & Wojewoda,(2015).
Understanding requirements	Tarawneh (2011); Jurison (1999); Procaccino, Verner, Overmyer & Darter (2002); Poon and Wagner (2001); Hussain & Mkpojiogu (2016)
Changes to requirements	Hastie & Wojewoda (2015); Serrador & Pinto (2015); Uskov (2017)
Quality control	Attarzadeh & Ow (2008); Wasserman (2010); Jureczko & Madeyski (2010); Xia et al. (2017)
Availability of subject matter experts	Castillo, Castro-González, Fernández-Caballero, Latorre, Pastor, Fernández-Sotos & Salichs (2016); McIntosh, Kamei, Adams and Hassan (2016)
Scope and size of projects	Alqahtani et al. (2014); Tarawneh (2011); Keil, Cule, Lyytinen & Schmidt (1998); Thakurta (2013)
Project management iron triangle	Atkinson (1999); Bronte-Stewart (2015); Kapczyński (2016); Neves, Borgman & Heier (2016); Lech (2013)
Risk management	Wallace et al. (2004); De Bakker, Boonstra & Wortmann (2010)
Methodology adopted	Tarawneh (2011); Chow & Cao, (2008); Verner and Cerpa (2005); Serrador and Pinto (2015); Khoza & Marnewick (2020)
User involvement	Bano & Zowghi (2015); Octavianus & Mursanto (2018); Tarawneh (2011)

2.6.1 Factors influencing software project success

Level of staff skills

The level of staff skills refers to whether staff members working in a software development environment have the necessary skills to do their jobs and ensure that projects will be successful. The skills that employees, project managers and developers need differ, depending on their role (Hastie & Wojewoda, 2015). In order for staff to be skilled, they require training and education. Training and education are necessary to allow individuals to continually develop their skills, which further allows them to become better at their jobs (Alqahtani et al., 2014). A skilled staff consists of people who understand what their roles are and what they are required to do to ensure the success of a project (Fui-Hoon, Lee-Shang & Kuang, 2001; Tarawneh, 2011). Octavianus and Mursanto (2018) attempted to rank factors based on how critical each of them was to project success. They found that skilled staff was ranked as the most critical success factor.

A study was conducted that investigated what the most important skills were for an IT project manager, what was the relative importance of those skills, and how important the skills of the top IT project manager were (Keil, Lee & Deng, 2013). Nineteen project managers were interviewed, with an average of 12 years experience. This study explained that project managers needed to have numerous skills to manage projects. These skills included leadership, verbal communication, scope management, listening, and project planning skills (Keil, Lee & Deng, 2013). Scope management appeared to be the most important because the scope of a project can change drastically during a project, which may further affect the schedule and cost (Keil, Lee & Deng, 2013).

Mtsweni, Hörne & van der Poll (2016) conducted a study to understand what skills were needed by various IT professionals. They found that project managers needed to be team players, conduct meetings, plan and control, have personal integrity, have listening skills, and have the ability to work in a group and work under pressure (Mtsweni, Hörne & van der Poll, 2016). Developers needed to be team players, have personal integrity, be able to work in a group, effectively manage their time, communicate and listen, be able to solve problems and think critically, be trustworthy and have the ability to work under pressure (Mtsweni, Hörne & van der Poll, 2016).

Employee experience

Most organisations hire employees based on their work experience, because there is a perception that experienced workers perform better (Dokko, Wilk & Rothbard, 2009). Project manager experience contributes to success. Their experience will guide the team in the best way to undertake a project, advising what can go wrong and how to avoid those issues (Tarawneh, 2011; Alqahtani et al., 2014). Project management expertise is the ability to apply knowledge, value and skills to manage a project effectively so that it adds value to a company (Hastie & Wojewoda, 2015). Project managers need to set out deadlines and milestones to meet all the objectives of the project (Alqahtani et al., 2014).

Experienced software developers may contribute to success as these developers understand how the architecture works and how the new features will fit into that architecture. Experience may be defined on the number of years a developer has worked in a place, or the number of projects they have worked on (Rehman, 2006). Peter Norvig of Google (2014) maintains that to be a talented programmer or project manager, one would need to practice 10 000 hours. If one works around 20 hours a week for ten years, that will equate to 10 000 hours.

Work-life balance

Employees need to have a work-life balance to focus on their jobs so that they can perform optimally. Overworked employees tend to make more mistakes, leading to project failure. Project success is therefore dependent on balanced time between work and other aspects of one's life such as hobbies, spending time with family and friends (Perera, 2011). In a study conducted to understand how work-life balance relates to job satisfaction, it was found that employees with a balance between their jobs and personal lives had a greater commitment to their organisations (Arif & Farooqi, 2014). This in turn meant that employees worked harder, thus producing good work and promoting project success (Arif & Farooqi, 2014; Heeks et al., 2017).

Pathak (2018) investigated the relationship between working hours, income and perceived organisational support and work life balance in banks in Nepal. The results found that a positive

relationship existed between organisational support and work life balance. This reveals the importance of organisational policies and value for management especially for married employees who have children (Pathak, 2018). A positive relationship also existed between income and work life balance indicating that the more income a person earns the more balanced their work and personal life is (Pathak, 2018). A negative relationship was found between working hours and work life balance. It was found that 82.4 percent of employees worked more than 8 hours a day. No work life balance existed as most employees worked more than they were required too (Pathak, 2018).

Job satisfaction

Job satisfaction can be defined as the extent to which employees feel content, satisfied and motivated in their jobs. An investigation was conducted to understand how job satisfaction and stress affects productivity (Halkos & Bousinakis, 2010). Some 425 employees were asked a series of questions that related to their jobs and how they felt at work. It was found that increased stress leads to reduced productivity and increased job satisfaction increases productivity (Halkos & Bousinakis, 2010).

Satisfied employees contribute to the success of a project by taking the initiative to do their jobs properly, communicating with each other effectively and creating a product that everyone can be proud of (Korrapati & Eedara, 2010). It is also worth noting that age and family status play a major role in job satisfaction, as the older the age of the employee, the less the ambition (Halkos & Bousinakis, 2010). It was also found that employees may be motivated by remuneration and benefits, which may lead to job satisfaction. However, Halkos & Bousinakis, (2010) found that satisfaction levels are not directly associated with remuneration.

Developers linked the success of their projects to job satisfaction, contentment with the working environment and how much they could learn and be creative during the process (Linberg, 1999).

Communication

Communication can be described as practices that increase the exchange of information within the development team (Ahimbisibwe et al., 2017). Communication assists teams by enhancing the collaboration and knowledge sharing and reducing team conflict which promotes a stable working environment (Ahimbisibwe et al., 2017). A case study investigated what team factors were vital to a software project (Estler, Nordio, Furia, Meyer & Schneider, 2014). It was found that communication was a key factor in project success and that 13 out of the 18 projects that were successful had weekly team meetings to communicate about the project (Estler et al., 2014).

Most of the communication took place via instant messaging as this is real time as opposed to having meetings every day (Estler et al., 2014). Communication occurred when team members needed help and when they needed to communicate on the status of the project. Communication is therefore necessary as it allows transparency amongst team members and allows for issues, if there are any, to be resolved more quickly (Estler et al., 2014). The need to communicate with the project team is vital for any software development project as everyone in the team needs to know how the project is going (Sheffield & Lemétayer, 2013). Communication via emails, meetings or even instant messaging is imperative for success as it ensures that the project is on the right track (Sheffield & Lemétayer, 2013).

Motivation

Motivation is what moves someone to do something (Takpuie & Tanner, 2016). Takpuie and Tanner (2016) classified motivation into a few categories, based on their research. Team members may be motivated by the enjoyment of doing their work, financial motivation and peer recognition (Takpuie, & Tanner, 2016). Team members need to be motivated to complete tasks and projects. Motivation can either come from management or other team members, or from realising self-goals (Daim et al., 2012). Estler, Nordio, Furia, Meyer & Schneider (2014) hypothesised that there is no difference in team motivation between agile and structured approaches. In their findings, they reported that no relationship existed between agile and structured approaches and team motivation. However, they did find that team motivation led to project success (Estler et al., 2014).

Equality

Employees that are treated fairly and are respected by their colleagues, as well as their managers, will produce better quality work as they feel that they are valuable to the company (Beecham, 2014). This directly affects success as employees want to maintain their level of respect by performing optimally, even getting rewards for their job performance (Beecham, 2014). Kivelä, (2019) investigated whether gender inequality affected software development project success. It was found that males and females were treated differently and this had a direct impact on the success of a project. It was found that if one gender was mistreated, it negatively affected a project, as the employees did not feel comfortable in their working environment.

Accountability

Accountability is a factor that relates to how the project is managed and this affects the success of a project. Everyone in the project team needs to be accountable for their own work, their role in the project, and the project itself (Reed & Knight, 2010). McHugh, Conboy and Lang (2011) investigated agile practices to understand what factors promote agile behaviour. Their research was based on three case studies that consisted of 27 participants. It was found that accountability amongst team members facilitated trust and promoted a greater work ethic, as well as respect for each other's abilities (McHugh, Conboy & Lang, 2011). Accountable employees limit team animosity and promote project success (Reed & Knight, 2010).

Support from management and stakeholders

Support from management and stakeholders is necessary for a successful project (Fui-Hoon et al., 2001; Tarawneh, 2011). Management and stakeholders can support, motivate and provide their team with resources, thus allowing team members to be motivated to work well, which leads to project success (Rezvani, Chang, Wiewiora, Ashkanasy, Jordan & Zolin, 2016). Executive support provides financial and emotional support by assisting and encouraging the successful completion of the project (Hastie & Wojewoda, 2015; Octavianus & Mursanto, 2018).

Dezdar and Ainin (2011) investigated what factors contributed to organisational impact. They sent out questionnaires to employees that built ERP systems and had a total of 384 respondents. The study showed that a positive relationship existed between support from management and

stakeholders and organisational impact (Dezdar & Ainin, 2011). It was concluded that this kind of support does affect project success and that the higher the support, the greater the chance for project success (Dezdar & Ainin, 2011).

Clearly defined business objectives

A sub-factor of organisational factors is clear business objectives. If business objectives are clear it will allow the project team to understand the project requirements earlier, ensuring the success of the project (Tarawneh, 2011). Clear business objectives provide an understanding of all the stakeholders' needs as well as the project requirements. They may also align with the organisation's goals and strategy (Hastie & Wojewoda, 2015).

Organisational culture

Organisational culture includes experiences, values, beliefs and behavioural norms. These factors influence the behaviour of individuals in an organisation and affect how decisions are made, who makes them, how individuals are treated and how the organisation responds to its environment (Tarawneh, 2011). Organisational culture plays an important role in project success as they shape employees' attitudes and practices (Eldrandaly et al., 2015). There are different organisational cultures that are adopted by organisations based on what is suited to them and what they believe will contribute to the success of a project. There have been conflicting opinions amongst researchers as to whether an agile organisational culture promotes project success (Ahimbisibwe et al., 2017). It has been argued that cultures that are more flexible and that are risk takers are more suited for an agile methodology (Ahimbisibwe et al., 2017).

Organisations need to understand what culture is best suited for to them and introduce it as routine activity in their employees' daily lives (Lee, Shiue & Chen 2016). Lee et al., (2016) investigated types of organisational cultures and what factors from each of them make them more suited to an organisation's software process implementation. The study showed the differences between a clan and hierarchy culture. A clan culture, which proved to be more successful, included a sense of flexibility that promoted team work, trust and increased knowledge sharing (Lee et al., 2016). A hierarchical culture was a more structured approach and needed formalized processes and procedures to facilitate communication and knowledge sharing (Lee et al., 2016). The study thus showed that an organisational culture must be carefully thought of in order to aid employee's productivity and software project success.

A study hypothesised an alignment between organisational culture and project performance (Gu, Hoffman, Cao & Schniederjans, 2014). In order to prove this hypothesis, data was collected from the United States of America and China. The results showed that organisational culture did have a slight positive impact on project performance (Gu, Hoffman, Cao & Schniederjans, 2014). The internal environment, such as the layout of the office, desk, proximity to other employees and cleanliness of the work space in which employees work affects their job performance, which further affected the project. When environmental pressures increased, employee's performance was negatively affected. Therefore, it was found that employees need to work in an environment that is not stress driven to ensure a positive project outcome (Gu, Hoffman, Cao & Schniederjans, 2014).

Tornjanski et al., (2015) describes banking as a very conservative industry which may be very resistant to change. Banking needed a stable environment and industry structure which needed clearly defined business models and defined boundaries which resulted in a slower pace for innovation as compared to other industries. However, in recent years, this culture has shifted as banks now drive digitization (Tornjanski et al., 2015).

Understanding requirements

Understanding requirements is important in all projects. Employees in software development teams need to understand the project requirements so that the correct product is built for the user. Requirements are gathered and documented differently in different organisations and work structures. Understanding requirements therefore means identifying the project scope, documenting the requirements and aligning these with the organisation's business objectives (Tarawneh, 2011; Jurison, 1999). If requirements are not analysed correctly it could lead to project failure (Procaccino, Verner, Overmyer & Darter, 2002).

Poon and Wagner (2001) conducted a case study in six organisations to identify the critical success factors in projects. They discovered that 50% of the interviewees regarded the requirements phase as vital to all projects as they need to understand what needs to be created. Developers in these organisations stated that they sit with users to fully understand what is required. They make executives and stakeholders go through these requirements, and arrange site visits to understand the daily lives of the users of the system (Poon & Wagner, 2001).

Hussain & Mkpojiogu (2016) found that requirements engineering was the foundation of every successful project. The success or failure of a software project is reliant on the accuracy, documentation and management of the requirements (Hussain & Mkpojiogu, 2016). It is also important to find a good technique for gathering and documenting requirements so that everyone understands the requirements (Hussain & Mkpojiogu, 2016). Requirements engineering must therefore occur at the start of every project and must be managed in a software or product development lifecycle to ensure success and mitigate against failure (Hussain & Mkpojiogu, 2016).

Changes to requirements

Changes to requirements can occur anytime during the project lifecycle, thus it is important that there are measures in place to accommodate these changes (Hastie & Wojewoda, 2015; Serrador & Pinto, 2015). Being adaptable to change is difficult, as employees cannot predict when requirements are going to change. Tools and models can be used to accommodate these changes. Using an agile approach attempts to cater for all changes requested by users during the project lifecycle, whereas in a traditional approach this is much more difficult. By holding meetings regularly to discuss the technicalities as well as the solution of a problem, and creating a 'whiteboard' change tracker, change management can become easier to deal with (Attarzadeh & Ow, 2008). Iterative development and prototyping, used in agile approaches, enhance the project development process. It allows users to test and provide feedback that further ensures that what the user wanted is being developed and that changes are easily adopted (Attarzadeh & Ow, 2008).

Uskov (2017) found that it was necessary to choose a methodology, such as the agile, scrum or Kanban approaches, that accepted changes. Projects will constantly require changes, and being able to accommodate changes will ensure the success of a project (Uskov, 2017). However, accepting changes will only produce successful projects if the correct methodology is adopted (Uskov, 2017). In agile projects it is necessary to welcome changes to the requirements, as this will ensure the success of a project (Gravell, Howard, & Aldahmash, 2017).

The traditional approach does not accept changes to requirements very well (Kramer, 2018). It is usually costly to make changes to the documentation and code; thus requirements are meant to

be defined accurately before the next phase of the project can begin (Kramer, 2018). If there are changes to requirements, they must be assessed to understand what the impact of these changes will be, before they can be made by the team (Kramer, 2018).

Quality control

The success or failure of a project is dependent on software quality control (Attarzadeh & Ow, 2008). Quality control refers to how good the software quality is. Testing is an important part of any software development process to ensure that good quality software is produced (Wasserman, 2010). Testers need to ensure that the software is safe, secure and works as it was specified too. Successful software quality control requires two phases, namely defect prevention and defect removal. Defect prevention produces good coding practices as well as a thorough analysis of requirements to minimise the possibility of errors. Defect removal requires activities at every stage of the development process to find errors and eliminate them (Attarzadeh & Ow, 2008). For example, it is not enough to test a mobile application that a customer is facing on an emulator; this should be tested on the devices that customers will be using to interact with the application (Wasserman, 2010).

Jureczko & Madeyski, (2010) stated that software testing is a tedious process that consumes time and resources; hence there is a need to create a model to predict and prevent software defects. In their attempts, they discovered that various models can be used to predict defects, based on the software project, the organisation, and the type of software that is being developed (Jureczko & Madeyski, 2010).

Xia, Lo, Bao, Sharma & Li (2017) found that the quality of software produced was critical to software project success. They concluded that if many defects are found in the product, it will create a bad user experience and perhaps even cause critical functionality to break whilst users attempt to use the software (Xia et al., 2017).

Availability of subject matter experts

A subject matter expert is someone who is skilled and knowledgeable in one particular field of study or practice. Subject matter experts in the software development environment are usually required to design a solution for the requirements and to review code to ensure that defect-prone code does not permeate into the product that is released to the customer. Subject matter experts are supposed to be involved in all aspects of a project as they play a transversal role in providing expert knowledge that is needed for the success of a project (Castillo, Castro-González, Fernández-Caballero, Latorre, Pastor, Fernández-Sotos & Salichs, 2016). Solution architects are required to design solutions, based on their knowledge which they turn into algorithms. Developers are then required to code the project based on the algorithm design (Castillo et al., 2016). Subject matter experts are also required to assess the performance of the system. Thus, the success of a project is highly dependent on the involvement and participation of subject matter experts (Castillo et al., 2016).

McIntosh, Kamei, Adams and Hassan (2016) hypothesised that the lack of subject matter experts in a project could result in software of poor quality being produced. An abundance of subject matter experts ensures project success as this ensures that every piece of code that is built is of quality, so that the end product exhibits good quality. Subject matter experts are therefore imperative to software development project success (McIntosh, Kamei, Adams & Hassan, 2016).

Scope and sizing of projects

The scope and size of a project establishes a clear vision, as well as measurable project objectives, for a project. It also prescribes strategies to achieve the vision and objectives. It identifies opportunities and limitations and clarifies the amount of work required (Alqahtani et al., 2014).

Minimising project scope allows for more achievable deliverables to be set and met (Tarawneh, 2011). A project can be split into phases to minimise the scope of a project. Each phase will be a different version of the software with more features added in each version. The scope of a project and reliable estimates go together, as the estimation is dependent on the size of the project (Keil, Cule, Lyytinen & Schmidt, 1998). A project must be sized to know exactly what the project will cost and how long it will take to complete (Tarawneh, 2011).

Scope creep can be defined as the addition of requirements during the execution of a project (Thakurta, 2013). The causes of scope creep usually stem from lack of comprehension of the project, the behaviour of the users or project members and actions or decisions from stakeholders (Thakurta, 2013). A study was conducted that investigated the effect of scope creep on a medium-sized project within an organisation that used the waterfall approach (Thakurta, 2013). It was found that an increase in scope creep directly increases the size of a project, thus putting pressure on the project schedule and team members (Thakurta, 2013). The increase in the scope and size of projects has a significant negative impact on project success.

Project management iron triangle

The project management iron triangle usually refers to a project being on time, within budget and meeting the requirements (Atkinson, 1999). This was traditionally used to determine if a project was successful (Bronte-Stewart, 2015). The iron triangle provides IT professionals with the ability to develop systems in a productive and well-planned manner that will further ensure the success of a system.

Kapczyński (2016) investigated what the critical success factors for a biometric authentication software project were. One of the factors that was identified in this study was that the project needed to comply with the project management iron triangle: so the project must be delivered on time, it must not run over the allocated budget and it must meet the requirements (Kapczyński, 2016).

The literature was reviewed to discover what traditional project management performance criteria are, and how the perceived project management performance differs from the project management iron triangle (Neves, Borgman & Heier, 2016). Traditional criteria referred to the project adhering to the schedule, functionality and cost (Neves et al., 2016). Respondents, managers and developers were interviewed to gain insight into how they perceived IT project management performance. Respondents believed that a project needed to respect the schedule and budget, which was necessary for good project performance (Neves et al., 2016). It was pointed out that, even though the project management iron triangle provides a guideline to a successful project, there will always be other factors that need to be measured as projects differ, and thus the way their success is measured can differ (Neves et al., 2016).

Empirical research to investigate whether projects being on time, within scope and within budget contributed to success was conducted (Lech, 2013). A survey was sent out to enterprises that developed software in Poland and a total of 28 enterprises responded (Lech, 2013). The success criteria were ranked from positions one to eight. The study showed that a project being on budget was a major success factor as it ranked at position one (Lech, 2013). Meeting requirements also plays a role in success as it tied for third position; and a project being on time ranked fifth (Lech, 2013).

Applying risk management

Risks need to be identified and controlled for all projects. Risk assessment must contain an analysis of all possible risks; their level of severity; what can be done to prevent each risk; how to deal with the risk if it does occur; and how to effectively reduce future risks for similar projects (Wallace et al., 2004). Once project managers have these risk assessments, they can take the necessary measures to prevent them from occurring (Wallace et al., 2004).

A study investigated whether risk management affects project success by reviewing numerous journals and articles (De Bakker, Boonstra & Wortmann, 2010). Different researchers had different ideas of what success means, and thus also different views on where risk management affects success (De Bakker et al., 2010). It was found that, of the various approaches to identifying risks, the main method should be creating a list of risks and ranking them in order of priority (De Bakker at al., 2010). Thereafter, discussions need to be held on how to manage these risks (De Bakker at al., 2010). This will ensure project success, as risks will be identified and mitigated.

Methodology adopted

The methodology that a team adopts guides how the project deadlines are set, what each team member is required to do and how the project will be managed. Methodologies can be structured, indicating that there are a set number of steps to follow; and each must be completed before the next. A structured methodology allows the team to follow a template and sets out a definite plan to follow (Tarawneh, 2011). Structured methodologies can sometimes be referred to as traditional approaches. Methodologies can also be flexible, which means that they welcome changes at any

point in the software development lifecycle. Flexible methodologies are referred to as agile approaches.

Different factors contribute to a successful project, depending on whether an agile or traditional approach is adopted. Adapting efficiently to changes in user requirements is a success factor that agile methodologies exhibit. The ability to cater for any changes during the course of the project allows for more features to be developed to produce a good product for customers (Chow & Cao, 2008). Agile methods have the ability to include more customer involvement, which allows companies to test out their prototypes more quickly and improve on their final product. This is considered a factor for success as more user involvement contributes directly to a better quality software product (Chow & Cao, 2008). Traditional methods structured approaches offer managers a better way to track a project, which can also be a success factor. They also require the organisation to complete each phase before starting the next. The structured and organised nature limits the uncertainty of a project, thus contributing to project success. Traditional approaches also work well with larger projects.

Verner and Cerpa (2005) investigated what development practices were used by Australian software practitioners. They discovered that 31.1% of the development teams used a waterfall approach, which allowed for the project managers to manage the project effectively (Verner & Cerpa, 2005). These projects showed project success, as strict methods were adhered to by the project manager to ensure that the rest of the team met the necessary deadlines and the requirements were met (Verner & Cerpa, 2005).

Serrador and Pinto (2015) asked 859 people about the success of 1386 projects. They discovered that agile and iterative methods were widely adopted as the preferred process for software projects. Out of the 1386 projects, six percent were completely agile while, 65% had some agile and iterative components that contributed to the project success (Serrador & Pinto, 2015). It was also found that the more agile a project is, the higher the success of the project (Serrador & Pinto, 2015).

Khoza & Marnewick, (2020) conducted a South African study in the attempt to compare which software development methodology leads to project success. The study measured success under five categories i.e. process, project management, deliverable, business and strategic success.

Process success measured how well teams select, integrate and implement process. It was found that an agile process was easier to implement than waterfall (Khoza & Marnewick 2020). When project management success was compared between the different methodologies, no notable difference was found, thus this construct was not dependent on the methodology. Deliverable success is based on the final product that's delivered (Khoza & Marnewick 2020). This construct showed agile was more successful as end users are involved during the entire project to ensure the correct requirements are delivered. Business success showed no difference when either methodology is adopted. The last construct measured strategic success which showed that agile had a bigger impact than waterfall based on the market and industry impact. This may be due to the fact that agile projects are released quicker to the market and the impact can be assessed sooner than waiting for a waterfall project to be released. The study concludes that adopting an agile approach is perceived to contribute to success (Khoza & Marnewick 2020). However, they also found that the waterfall projects in the sample were much larger and lasted longer than the agile projects. The waterfall projects spent 10.16% less than budget, but Agile spent 42.62% more than budget; both spent more time than planned, but waterfall spent less time (13.65%) than Agile projects did (22.4%) (Khoza & Marnewick 2020).

User involvement

User involvement is a critical success factor in any software project being developed, as users usually define what they want in a system (Bano & Zowghi, 2015; Octavianus & Mursanto, 2018). Frequent user involvement will allow the project team to anticipate changes, obtain feedback and improve on functionality (Tarawneh, 2011).

In a systematic literature review of journals and articles that investigated the relationship between user involvement and system success, a positive correlation was found (Bano & Zowghi, 2015). Out of the 87 empirical studies that were examined, 52 studies stated that user involvement contributed to project success. However, 12 studies stated that user involvement negatively impacted success and 23 studies were uncertain (Bano & Zowghi, 2015).

2.6.2 Project managers' perspectives of software development project success

The opinions of what success means to different people may depend on the role that they are in, the types of projects they are given and the team that they work with. A discussion follows on what project managers think project success is.

Verner and Evanco (2005) surveyed 101 respondents about 122 projects in an attempt to investigate what project management practices lead to success. It was found that five percent of their sample did not have a project manager on their projects as most of them were small. The small projects had fewer than seven people working on them. The larger projects had project managers with between six months and 22 years experience (Verner & Evanco, 2005). The research showed that a project manager is needed to ensure the success of a project (Verner & Evanco, 2005). A project manager that understands the project goals and requirements, plans deadlines effectively, and manages costs, will ensure project success (Verner & Evanco, 2005).

Wateridge's 1998 study investigated project success by interviewing project managers, sponsors and users to find the five most important criteria. It was found that meeting user requirements was the most important criterion according to the project managers and users. Users said that user happiness with the product is also one of the main factors of success, whereas project managers were more inclined to choose meeting deadlines and sticking to the cost estimates (Wateridge, 1998). It was clear that users deemed themselves important and project managers were concerned with their jobs and satisfying their bosses (Wateridge, 1998).

Project outcomes are affected by the people, and the technical and functional aspects of project management (Hughes, Rana & Simintiras, 2017). Project success is therefore linked directly with project managers and their ability to effectively deal with people (Hughes et al., 2017). Project managers measure success by projects being on time, within budget, meeting user requirements and meeting stakeholder expectations.

2.6.3 Software developers' perspective of software development project success

Software professionals, project managers and developer, have different perceptions of project success. It was found that, even though project managers may regard a project as unsuccessful

because it is over budget, over schedule and over scope, developers may still deem it successful (Linberg, 1999). This success is perceived because the software worked the way it was intended to and met all requirements (Linberg, 1999). Developers regarded the success of a completed project as the quality of software produced. Even projects that were cancelled were deemed a success by developers if they could learn from them (Linberg, 1999). From the developers' perspectives, the technical quality of the software is more important than the organisational impact.

In another study, developers were found to think of success in two categories: the personal factor, which gives them a sense of achievement by completing a job; and the customer factor, that allows users to be involved in the development process in order for the team to develop the correct requirements (Verner & Evanco, 2003). Meeting the project schedule or staying within budget are not considered factors of success by developers (Verner & Evanco, 2003). Some of the other factors that software developers consider as success factors include whether management is involved or not; whether requirements are properly defined; if project management is conducted effectively; whether the development process exhibits any environmental issues such as setting up their computers or loss of team members; and the ability to cater for changing requirements from users (Verner & Evanco, 2003).

Some researchers have created models to help them understand how to measure project success and to predict project outcomes (Misirli, Verner, Markkula & Oivo, 2014). Projects should not only be considered successful if they have met the desired timelines, cost and quality criteria (Misirli et al., 2014). Software developers are affected by their ability to perform in their job. If developers are able to learn new skills that allow them to apply these on future projects or jobs, they consider a project successful (Misirli et al., 2014). The fact that developers can be creative and complete challenging projects motivates them to perform well, which further allows them to believe that the project was successful (Misirli et al., 2014).

From the above literature, it can be seen that software developers regard project success as being able to solve problems creatively; produce good quality software and learn new skills. Project managers and software developers clearly have conflicting opinions as to what project success means. It is evident that project managers are more business orientated whilst software developers are more technical. The project manager's main focus is on meeting the business

objectives, whereas developers are guided by meeting the requirements that were specified to them.

2.6.4 Software development project success in financial institutions

Software development is critical to any industry as it is needed to automate and streamline processes, either to improve customer experience or internal business processes. The banking industry also uses software to do this. It is important for banks to constantly improve on internal and customer facing systems to reduce employee and customer angst. To our knowledge, there are not many studies that investigate software development project success in banks.

Robinson and Sharp (2005) investigated organisational culture and extreme programming with three different cases studies. The first case study was in a multinational bank. The bank exhibited a hierarchical organisational culture. The XP team was divided into two sub-groups and each of the groups had a developer who was the team leader (Robinson & Sharp, 2005). The bank had strict policies and procedures that governed how employees worked, dressed and interacted. The overall working environment consisted of an open plan office with workstations set up in rows (Robinson & Sharp, 2005). Projects were completed on time and met the requirements. However, due to the strict nature of the working environment, employees were not satisfied.

In their study on project management practices that lead to success, Verner and Evanco (2005) distributed their questionnaires to financial institutions, pharmaceutical companies, insurance companies and banks. Their findings showed that success was dependent on whether the project manager had a clear vision of the project, whether requirements were well understood, and whether the timelines were achievable and met (Verner & Evanco, 2005).

A study by Heeks, Krishna, Nicholson and Sahay (2017) investigated the global outsourcing of software development for a US bank. This research revealed that the US bank gave their vendors smaller projects in an attempt to work in an agile manner (Heeks et al., 2017). A notable factor that was explored was work-life balance. It was found that employees rarely took vacation days, tended to work longer hours than were required and also sometimes worked on weekends. The reasoning behind this was that customers constantly required changes; thus employees worked

longer hours to meet these requirements (Heeks et al., 2017). It is clear that this is a stressful working environment which impacts employees' work-life balance, which may further impact project success (Heeks et al., 2017).

Akgün, Keskin, Ayar and Okunakol (2017) conducted a case study to understand how knowledge is shared amongst members in software development teams. Eighteen project managers from three Turkish banks were interviewed. During the interviews, the project managers were asked about the meaning of knowledge sharing; what kind of knowledge was shared amongst the team; the reason why they might not share knowledge; as well as suggestions to overcome the reluctance of knowledge sharing (Akgün et al., 2017). The results indicated that team members' reluctance to share knowledge was based on individual barriers as well as organisational barriers. Team members tended to use knowledge for controlling a situation as well as a means of defence. They also believed that their knowledge would guarantee their job, status and career (Akgün et al., 2017). Another factor that impeded the knowledge sharing process was the lack of reward for sharing information and knowledge (Akgün et al., 2017). Organisational culture also appears to have been an obstacle in the knowledge sharing process. Hostility by team members towards other team members and managers also contributed to the lack of knowledge sharing (Akgün et al., 2017). This research suggests that a lack of knowledge sharing or lack of communication between team members may cause project failure as team members will not have context when problems occur, which may delay a project (Akgün et al., 2017). A few suggestions were made to remedy the issues experienced by project managers when it came to knowledge sharing and communication (Akgün et al., 2017). Management needs to integrate the knowledge sharing process into employees' jobs. Management should also make use of knowledge sharing tools to cater for the documentation of project specific information that can be shared with the team (Akgün et al., 2017). Management must ensure that their employees feel safe and secure in their roles, so as to leverage knowledge sharing motivations (Akgün et al., 2017).

Shahibi, Sarifudin, Hussin, Ibrahim, Ali & Fakeh, (2019) investigated project success through the project management iron triangle factors including quality and people management in a financial institute in Malaysia. The population for their study included project managers who had worked on over 150 software projects. Their proposed model showed that the project management iron triangle inclusive of people and quality did offer a strong framework to measure project success (Shahibi et al., 2019). All their constructs produced means higher than 3.5 out of a maximum of 5 indicating that there was a strong inclination that these factors led to a successful project. Shahibi et al. (2019) found that the project management process is complex and requires

extensive attention beyond human, budget and technical variables. Whilst these factors worked in this financial institute, each project may have different factors thus designing the success measures around each project will ensure that it will be successful.

There has been limited investigation into software development project success within banks. This study will therefore attempt to contribute to the body of knowledge of how the software development process is experienced by project managers and software developers within a South African bank.

2.7 Software development project failure

The definition of software development failure is problematic as it is perceived to be vague and difficult to measure (Lehtinen, Mäntylä, Vanhanen, Itkonen & Lassenius, 2014). Failure can be associated, either with the process of developing software, or with the outcome of what is produced (Lehtinen et al., 2014). The Standish Group defined software development project failure as a project being cancelled or not meeting business objectives, not being within budget or not being on time (Hastie & Wojewoda, 2015). The Standish Group found that on average, 19% of projects fail (Hastie & Wojewoda, 2015). When a software project fails, it jeopardises an organisation's reputation and future prospects (Charette, 2005). Whilst this may be true for most projects, there are other factors that influence software development project failure. A few researchers have investigated software development project failure. A discussion follows.

Projects may fail as a result of a combination of factors and a number of factors must be evident before a project can be deemed as a failure. Verner, Sampson and Cerpa (2008) analysed data from 70 failed projects. From their findings it was noted that the median number of failure factors was 28. The minimum number of factors that causes a project to fail is five and the maximum was 47 (Verner et al., 2008). Some of the main factors were that the due date had an impact on the delivery process; the project was underestimated in terms of timelines and scope; there was a lack of risk management; staff were not rewarded for working long hours; requirements were not analysed thoroughly and staff had a bad experience working on the project (Verner et al., 2008). It was concluded that project failure was a result of multiple factors. Based on the critical success

factors presented in section 2.6, a discussion follows below on how each of these factors will affect project failure.

2.7.1 Factors influencing software development failure

Staff skills

The lack of, or inadequate training of staff is seen as a contributory factor to failure. All employees constantly need to improve on their skill set as the environment that they work in is technology driven (Hughes et al., 2016). Technology is constantly changing; thus improving on skills ensures that software is developed according to industry standards, and that new and better ways to develop and manage software are applied (Hughes et al., 2016).

Employee experience

Whilst prior work experience is directly related to knowledge, employees may develop habits and behaviours from their previous jobs or projects that may not be useful (Dokko, Wilk & Rothbard, 2009). Lack of experience in using a specific methodology, or a lack of experience in the field of software development, can lead to project failure. Employees' lack of experience in the field or in the process results in incorrect software being produced and deadlines not being met (Marques et al., 2017). Assigning complex or critical system projects to new employees can jeopardise a project as they will have no knowledge of how to go about completing and managing the project.

Work-life balance

Employees require a work-life balance to ensure that they do not 'burn out' at work. They need time to spend with their families and friends and doing things that give them joy (Johri, 2011). A lack of this balance results in employees spending more time at work, working longer hours and being dissatisfied in their jobs. This may lead to project failure as employees no longer feel a sense of pride in their work, and they can do their jobs badly (Johri, 2011).

Job satisfaction

Project failure may be influenced by job satisfaction in that staff loses commitment to a project if there is no motivation from the team, managers and stakeholders. In technology-based industries, employees often work longer hours to meet deadlines and solve complex problems to deliver software in unrealistic timelines and budgets. This may lead to increased stress levels which lower job satisfaction and performance (Hughes et al., 2016).

Team dynamics

A lack of communication in the team is one of the main factors that leads to project failure. Team members need to communicate the status of projects. Failure do so results in others not knowing what is going on with a project, and thus they cannot assist if there are issues. If an employee leaves that project or business and has not communicated how the project works, it could negatively impact the project (Marques, Costa, Silva & Gonçalves, 2017). When some team members are favoured over others, it can lead to project failure as they are not treated fairly. This decreases their work ethic, thus decreasing their job performance, which causes the projects that they are working on to suffer.

Support from management and stakeholders

A software development project can be negatively impacted if there is a lack of support from management and stakeholders. The lack of engagement and commitment from management and stakeholders further emphasises that the project is not urgent or important. This leaves employees with no sense of urgency; thus the project may be developed poorly, or requirements may be left out (Hughes, Dwivedi, Rana & Simintiras, 2016).

Clearly defined business objectives

The business case for any project must be aligned to the vision of the business – thus the business objectives need to be clearly defined (Yeoh & Koronios, 2010). If these are not understood, it will impact the outcome of the analysis of the requirements and the software being developed.

Business objectives need to be defined upfront to ensure everyone understands the goal of the project to prevent project failure (Yeoh & Koronios, 2010).

Organisational culture

Employees need to be comfortable in their working environment to produce good quality work. Organisational culture refers to the working environment – the people as well as the actual environment. When an environment is negatively impacted by outside influences, it can further affect the organisational culture. Employees forced to work in environments that are hostile will produce poor quality work, thus leading to project failure (Mishra & Mishra, 2011).

Understanding requirements

The inadequate definition of software requirements directly relates to project failure as this means that employees do not understand what is required from them and testers cannot properly test the new feature (Hughes et al., 2016). This factor is also impacted by the scope and sizing of projects, as large projects may introduce complexities that may not be documented well enough to understand (Hughes et al., 2016).

Changes to requirements

Software development projects can fail because of changes to requirements during the project lifecycle. An organisation and the project timeline need to cater for these changes to prevent the project from failing. Managing changes to requirements is an extremely difficult process that requires competency from employees and changes to the adopted methodology. This may result in failure if not accommodated earlier in the project (Hughes et al., 2016).

Quality of software

The quality of software is checked when the project goes into testing. Software testing is critical to any project as any new feature needs to be thoroughly tested before going live to customers. Lack of testing or inadequate testing could result in project failure as customers may be given a product that is incomplete or does not meet their expectations. Emphasis is therefore placed on

testing and defect management (Lehtinen et al., 2014). The lack of time for appropriate testing and rectifying of defects ultimately leads to project failure (Marques et al., 2017).

Availability of subject matter experts

A lack of competent software professionals results in failure. These subject matter experts are often preoccupied and cannot advise on solutions for other projects which can cause a project to fail (Jørgensen, 2016). Staff needs to be knowledgeable in all aspects of the business and system requirements which will limit the dependency on subject matter experts.

Scope and sizing of projects

The scope and sizing of projects can contribute to project failure. This is as a result of project timelines being underestimated, based on the size of the project (Nasir & Sahibuddin, 2011). Not all projects can follow the same timelines and larger projects may need more time and resources allocated. Projects that are large usually offer high levels of complexity which directly relates to failure most of the time (Hughes et al., 2016).

Project management iron triangle

A project may be on time, on budget and meet the defined scope but may not be successful. This is because even though, from a management perspective, the project met the project management iron triangle factors, failure may have occurred in other aspects (Savolainen, Ahonen & Richardson 2012). These aspects may have included poor quality software being produced or individual factors being affected that resulted in disgruntled employees.

Applying risk management

Poorly conducted risk management will lead to project failure. If the risks identified are not mitigated and controlled, they may cause further risks in the project (De Bakker, Boonstra & Wortmann, 2010). Not assessing risks, poor risk assessment and inaccurate risk estimations during the project lifecycle can also impose further delays on the project; thus timelines may not

be met (De Bakker, Boonstra & Wortmann, 2010). One of the main reasons for software project failure is that risks were not identified and mitigated at the start of the project (Aloini, Dulmin & Mininno, 2007).

Methodology adopted

The software development methodology must be chosen to suit the requirements of a project. A clear methodology that is understood by everyone needs to be selected to prevent project failure (Lehtinen et al., 2014). This is necessary as an approach needs to be followed in order for the project to be managed effectively. Based on the above, there is a clear dependency on factors. A relationship is noticed that each factor either positively or negatively influences the other (Lehtinen et al., 2014).

User involvement

Products are developed for customers who can be referred to as end users. Requirements are usually driven by what users need to make their lives easier (Nasir & Sahibuddin, 2011). User involvement is imperative to software development – during the requirements gathering process as well as the testing process. A lack of user involvement could result in a completely different product being developed which will then require further reworking. This wastes time and resources and impacts costs (Nasir & Sahibuddin, 2011).

2.7.2 Project managers' perspectives of software development project failure

Several factors influence software project failure, from a project manager's perspective. The first factor is the lack of top management support. Employees tend to give priority to, and focus on, what management deems as important. Management does not support all projects; thus employees will not spend time on smaller projects in which management is not interested, and which could potentially fail (Kappelman, McKeeman & Zhang, 2006). Support, involvement and motivation from management create a more stable and productive work environment. A lack of this support and motivation, therefore, may influence a project negatively (Whittaker, 1999).

Secondly, the lack of proper project management negatively affects project success. If project managers cannot manage a project and effectively communicate with team members, project failure will be definite (Kappelman et al., 2006). Inadequate project planning and management results in project failure as risks are not identified and therefore not eliminated (Savolainen, Ahonen & Richardson, 2012).

The third factor is the lack of stakeholder involvement. If a project has stakeholders, they will need to be actively involved as they will be required to provide resources in order for the project to progress. If they do not, it will halt the project indefinitely, and the project may therefore be unsuccessful (Kappelman et al., 2006).

Another factor is unskilled or unhappy project team members. Some team members may not produce good quality work and they may also not meet the deadlines. This affects the next person's work as their job may be dependent on their team members (Kappelman et al., 2006).

The last factor is the lack of subject matter experts. Organisations tend only to hire a few subject matter experts as they are rare and highly sought after. This results in subject matter experts being overscheduled as they will be required to consult with business units on all projects. The possible outcome of having only a few subject matter experts could be project failure as the subject matter expert is overwhelmed and unable to devote enough time to every single project (Kappelman et al., 2006). This may further result in the incorrect requirements being specified and can cause a miscalculation of project estimates which may lead to project failure (Whittaker, 1999).

2.7.3 Software developers' perspectives of software development project failure

Software developers know why projects have failed. Firstly, the underestimation of the due dates of projects results in the team not being able to conduct reviews at the end of each phase in order to eliminate issues within the project (Cerpa & Verner, 2009). This may result in risks that are not reassessed and controlled. This limits insight into a project, as team members and managers will be unaware of problems, such as developers having problems with the project (Verner et al., 2008).

A second reason is that risks are not assessed. When risks are not assessed, it is unclear what problems may occur. If they are resource allocation or requirement specific problems, the team may have a negative working experience (Cerpa & Verner, 2009). This further affects the team's motivation as an aggressive schedule may become necessary, causing the team to work long hours (Cerpa & Verner, 2009).

Another reason is that the project scope can be underestimated. When a customer does not provide a full set of requirements, the estimating process is affected as one cannot estimate what one does not know or understand what needs to be developed. This estimating process then affects staff and resource allocation, which further affects the schedule, development process, team motivation and team members' lives (Verner et al., 2008). Adding staff members later on in the project does not solve the problem as Brooks's law states that: 'Adding manpower to a late software project makes it later' (Brooks, 1975).

Lastly, developers believe that if the team has not been rewarded for working long hours, it can lead to project failure. Developers would feel undervalued, resulting in poor quality software being produced (Cerpa & Verner, 2009).

2.7.4 Prevention of software development project failure

Preventing software project failure is the main goal of any company as it saves on costs, may shorten delivery timelines and produces a quality piece of software to end users (Lehtinen et al., 2014).

Projects that have complex logic or involve software and coding practices that are unknown are best given to employees with experience. This prevents failure, as experienced employees will have a better understanding of how to implement and manage the project (Stellman & Greene, 2005). Staff members with different levels of skills can be allocated effectively to projects to ensure project success. All staff members may be skilled in different aspects and they can assist on projects in their own way. Ensuring that staff skills are utilised prevents failure, as everyone will have a task to complete which they can do efficiently (Stellman & Greene, 2005).

Software project failure can be prevented using a few techniques. It is important for a project to be correctly scoped and sized, given the timelines. Scoping and sizing projects correctly prevent project failure, as all requirements can be catered for and all deadlines can be met. A crucial step in any project development life cycle is requirements gathering. Analysing and documenting requirements is necessary in any project as it ensures that all requirements are catered for (Kaur & Sengupta, 2013). Ensuring that all project team members are part of this phase is vital. An example is having the technical resources, such as the developers and architects, advise on what is possible from a systematic perspective, as this prevents changes later on in the project (Kaur & Sengupta, 2013). Requirements that are documented clearly help the project team understand what is required from them so that they work confidently.

Requirements that are well documented help prevent changes later on in the project development life cycle. However, changes could come from management, stakeholders or even users. A technique to prevent failure is to make advance provision for changes in the project life cycle. This will ensure that, if any changes come in, they can be dealt with immediately without delaying or causing harm to the project.

Projects need to be well managed to prevent failure. A valid software methodology must be chosen and agreed upon prior to starting a project. This guides how the software development life cycle will unfold and what deadlines or milestones to work towards (Dorsey, 2000). The chosen software development methodology must ensure that sessions with the project team are held on a regular basis as this helps the team understand the status of the project (Dorsey, 2000). Knowing how a project is doing helps prevent failure as everyone can help on the outstanding issues, which results in the project being completed more quickly.

A defect can be thought of as the inaccurate flow of software, or a blemish in the product or process. Defect prevention is a vital part of any project (Suma & Nair, 2010). Defect prevention involves identifying the root causes of defects and how to stop them from reoccurring in the future.

There are a few types of testing that should occur to prevent project failure (Suma & Nair, 2010). These include regression, performance and environmental testing. Regression testing ensures that the new software does not impact the existing software negatively. Performance testing is

conducted to ensure that the new software works optimally by responding in a timely manner to a user (Suma & Nair, 2010). Environment testing is when testing is done in the environment that the user will be using. Testing of software is imperative to prevent failure, as testing ensures that anything that is wrong with the software is picked up and rectified before it is released to the user (Suma & Nair, 2010). Good software quality and coding practices reduce the number of defects found in the software, thus preventing software project failure. Developers need to work closely with their software architects to understand how to deliver good coding practices. This will ensure that the code is reusable and easy to understand by everyone (Lyu, 2007).

Management and resources need to be actively involved in projects to prevent failure as they can provide support and motivation when needed. Mangers and stakeholders need to ensure that they show an interest in all projects, as disinterest will indicate that the project is not important, and no sense of urgency will be accorded to it (Stellman & Greene, 2005).

Business objectives help employees understand what is required of them. If these are not defined, it can lead to failure. By clearly defining the business objectives and sending them out to the project team, it prevents failure as everyone has sight of what is required. Aligning to those business objectives is significant for the organisation as it drives their company's goals and vision (Stellman & Greene, 2005).

Ensuring that all employees have a comfortable working environment is essential to prevent project failure. Employees work and thrive better in environments where they feel safe, respected and motivated (Stellman & Greene, 2005). A noteworthy failure prevention technique is for managers to ensure that their employees are happy by having sessions with them to understand how they feel about their jobs.

Software failure can be prevented by numerous techniques. This is dependent on the project, the team and the organisation. Different companies apply prevention techniques that are best suited to them to ensure project success.

2.8 Conceptual framework

The above literature review provides an understanding of what factors contribute towards software project success. For this study, the main factors that were discussed in the literature review will be used to create a conceptual framework. The factors were chosen based on their relevance to the research topic, as well as their relevance to the organisation that is being researched and observed by the researcher. This conceptual model is grouped into individual factors, team factors, organisational factors and project factors. Figure 2 provides an overall view of the proposed software project success model for this study. The factors that have been identified will allow for an in-depth investigation into what makes a software project successful from a project manager's and software developer's perspective. An explanation of the factors follows below.

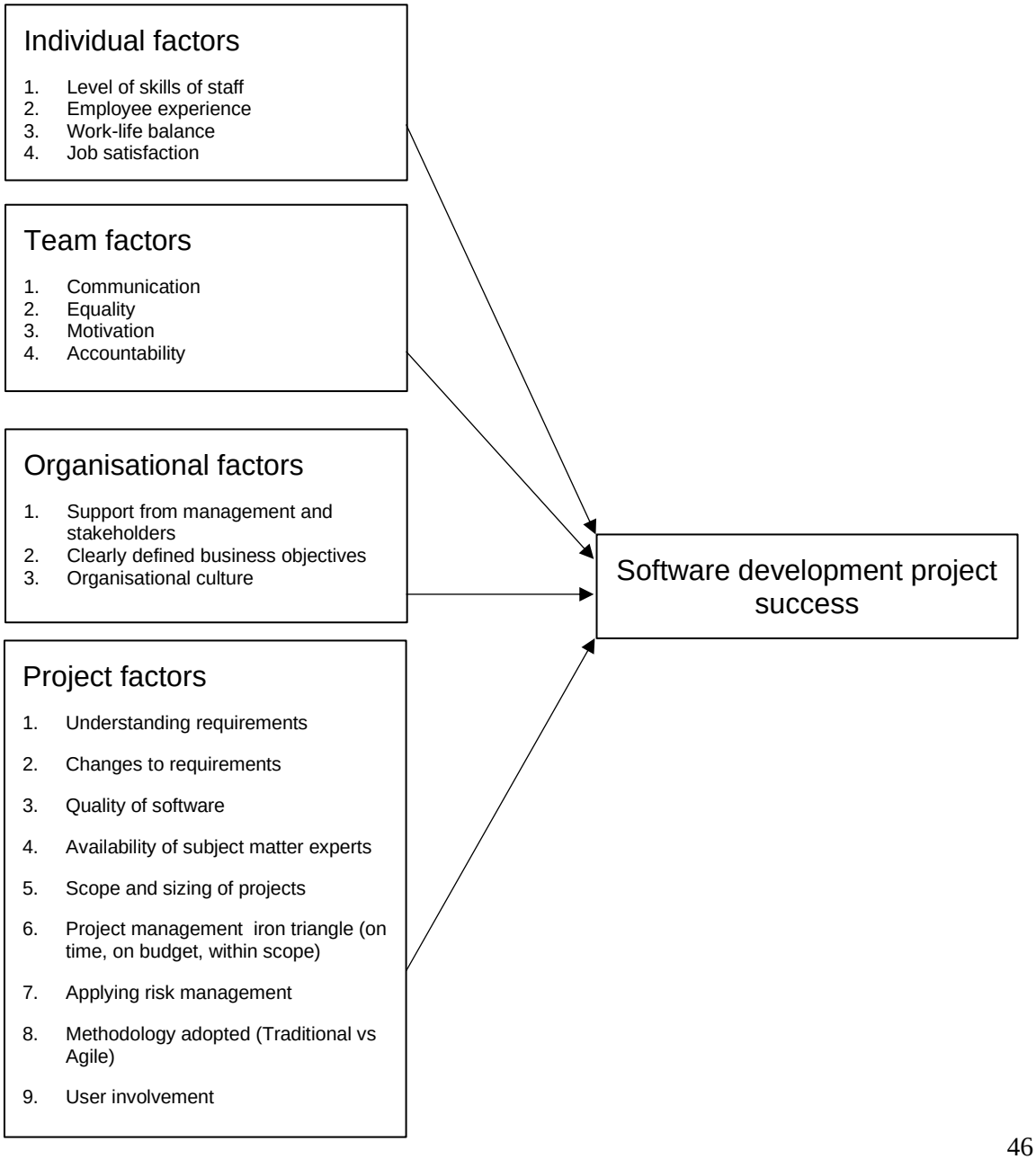

Figure 2: Proposed software project success model with factors and sub factors.

2.9.1 Individual factors

2.9.1.1 Level of staff skills

There are factors that can be placed in the individual category. The extent to which employees are skilled is dependent on them being given the time and resources to gain new skills; their experience and practice. If employees are constantly given projects, they will have a better understanding of what to do (Hastie & Wojewoda, 2015). As a result, they will tend to be allocated more projects than more novice employees. It is important to ensure that all employees have a minimum of skills and abilities to be able to complete any size of project (Fui-Hoon et al., 2001; Tarawneh, 2011). A lack of skilled employees could potentially lead to project failure as no one will be equipped to handle bigger projects. These bigger projects may have to be cancelled.

2.9.1.2 Employee experience

Employee experience is a second factor that affects success. Experienced employees are familiar with problems or issues that may occur, and as a result cater for these or prevent them from happening altogether (Alqahtani et al., 2014). Investigating how experienced employees are, will determine what level of experience employees should have, as well as how project success depends on their experience.

2.9.1.3 Work-life balance

It is important that employees have a work-life balance. An average work week in software development is 40 hours per week (Grossman, Bergin, Leip, Merritt, & Gotel, 2004). The ideal is that employees work at a comfortable pace, so that when they leave at the end of the day they do not feel stressed or tired and can come back the next day feeling refreshed (Grossman et al., 2004). Overworked and stressed employees are more likely to make mistakes that could lead to project failure (Grossman et al., 2004). It is therefore up to management, as well as each employee, to make sure that they have a balance between work and their normal lives. Work-life balance is therefore the third factor in the individual category.

2.9.1.4 Job satisfaction

Finally, job satisfaction can be derived from the ability to think of creative solutions to solve problems, as well the sense of achievement when completing a task (Linberg, 1999). Developers believe that this ability relates to successful projects where they can propose a solution based on their expertise (Crowston, Annabi & Howison, 2003). It allows developers and project managers to use their skills in a creative manner, which essentially provides them with job satisfaction. This factor will be investigated to determine whether project success is reliant on job satisfaction.

2.9.2 Team factors

The team factor consists of four sub factors. The dynamic between team members sets the way teams communicate and work together. The better the team dynamic, the easier it is for team members to work together (Guinan, Cooprider & Faraj, 1998). This leads to project success as team members will share knowledge easily and assist each other when challenges occur, which further reduces and prevents project-related issues from occurring (Guinan et al., 1998). Facilitating communication between team members is vital to any project (Cockburn & Highsmith, 2001). Team members need to communicate by talking about the status of a project, what is required from each other and what problems have occurred (Layman, Williams, Damian & Bures, 2006). This will help in the handing over of work to other team members by a colleague if he suddenly leaves; and everyone in the team is able to assist if problems occur (Cockburn & Highsmith, 2001).

2.9.3 Organisational factors

2.9.3.1 Support from management and stakeholders

The first factor is support from management and stakeholders. Support from top management and stakeholders means that they are involved in the project, from offering support and motivation to providing resources when required (Alqahtani et al., 2014). The lack of support of top management and stakeholders can lead to project failure as staff will not be motivated and may lack resources to complete a project (Alqahtani et al., 2014; Hastie & Wojewoda, 2015). Support from management and stakeholders is a factor in this investigation, as it is necessary to determine the level of support that management provides, as well as their involvement during the software

development life cycle (Fui-Hoon et al., 2001; Tarawneh, 2011). This further indicates that it may play an important factor in project success.

2.9.3.2 Clearly defined business objectives

The next factor is clearly defined business objectives. The definition of clear business objectives provides a project team with the ability to understand what is required of them from a business perspective. This allows them to work towards meeting these objectives, which results in project success (Hastie & Wojewoda, 2015). This factor will provide details as to how objectives are identified and met.

2.9.3.3 Organisational culture

The third factor is organisational culture. Organisational culture will determine the beliefs, behavioural norms, experiences and employee values within the organisation, as well as how decisions are made and how employees respond to their work environment (Alqahtani et al., 2014; Tarawneh, 2011). This factor affects project success as it determines the well-being of the employee as well as how they respond to their working environment. If employees dislike their working environment, they will not be as productive as if they enjoy working in this environment; thus project failure may occur (Alqahtani et al., 2014). An employee's values are important as it drives who they are as individuals. An acknowledgement and acceptance of their values by colleagues will ensure that they find their working environment safe and comfortable. The working environment should allow them to do their work without fear of being judged or being misunderstood. This will increase productivity and project success (Alqahtani et al., 2014).

2.9.4 Project factors

2.9.4.1 Understanding requirements

Understanding requirements is one of the most important parts of the software development process (Tarawneh, 2011). If requirements are not understood, the project team will not know what they are developing and project managers will make inaccurate estimates (Jurison, 1999). Requirements are provided by the customer and then translated by the business and system analysts to determine how these objectives meet the company's goals as a whole, as well as fitting

into the architecture of the system (Procaccino et al., 2002). Misunderstood requirements, therefore, have a ripple effect as they will affect the entire project development process (Tarawneh, 2011). The extent to which requirements are understood will provide insight into a company's software development process, as well as how they affect projects.

2.9.4.2 Change to requirements

The second factor is the ability of the software development process to accommodate changes to requirements. It is also referred to as change management (Cockburn & Highsmith, 2001). Traditional methodologies lack this ability, as requirements which change during the course of the project are not catered for. Agile methods, however, thrive on change, so an agile project will be able to accommodate changes in requirements (Serrador & Pinto, 2015). Requirements may change, which may further affect processes, whether the methodology is traditional or agile. Therefore, there needs to be some kind of change management to ensure project success. This factor will be investigated to understand how changing requirements affect the success of a project.

2.9.4.3 Quality of software

The third factor is quality of software. The reason why success or failure is dependent on quality of software is because finding and fixing any software defects is the most tedious and time consuming task of software development (Attarzadeh & Ow, 2008). An investigation into how software quality affects project success will be carried out. It will also provide information as to what aspects of quality are important.

2.9.4.4 Availability of subject matter experts

The fourth factor is the availability of subject matter experts. This will be explored to determine if the team has access to enough subject matter experts (Kappelman et al., 2006). This will also provide insight into whether the lack of subject matter experts affects project success.

2.9.4.5 Scope and sizing of projects

The size of a project may influence project success as it determines the amount of work required and affects resource allocation (Tarawneh, 2011). If a project is too big, recommendations and risks need to be considered as to whether this project should be done in phases (Keil, Cule, Lyytinen & Schmidt, 1998). Scope creep may influence a project negatively. Scope and sizing therefore need to be effectively managed (Fui-Hoon et al., 2001; Tarawneh, 2011). A misjudgement of these estimates may cause delays in projects as more resources may be needed, which could result in the project being cancelled (Tarawneh, 2011). Reliable estimates and strategic planning require that measurable goals, timelines and milestones must be identified clearly (Tarawneh, 2011; Alqahtani et al., 2014). Project sizing will provide details about how projects of different sizes and scopes are dealt with, and whether deadlines differ, based on these in the South African bank.

2.9.4.6 Project management iron triangle

The project management iron triangle will be investigated as the sixth team factor. It contains three sub factors. This will determine whether the extent to which a project is on time, within budget and within scope affects software development project success (Atkinson, 1999; Hastie & Wojewoda, 2015). It allows project managers to manage the expectations of the customer and the project team. Estimates of how long the project will take, what it will cost, and how it will affect the organisation, need to be conducted for all projects. This factor will provide insight into whether the project management iron triangle affects success.

2.9.4.7 Risk management

Risk can be defined as exposure to harm. Risk management, therefore, aims to prevent risks from occurring so as to ensure the success of a project (Wallace, Keil & Rai, 2004). An investigation into how risks are managed and controlled will be undertaken. Project managers and software developers must take precautions to prevent risks. It is therefore important to identify and control these risks in the form of risk assessments (Attarzadeh & Ow, 2008). This factor will provide an understanding of how these occur and what impact they have on project success.

2.9.4.8 Methodology adopted

The methodology adopted refers to whether the software development project follows a traditional or agile process (Tarawneh, 2011). This factor contains two sub factors. Within both of these methodologies there are models that further define what process is used. This factor affects project success as it provides a basic guideline that a project should follow. It provides team members with a process, the phases of the project, what is required in each phase, and deadlines; and it allocates tasks and manages expectations (Chow & Cao, 2008). Methodology will be examined in this study to uncover whether an agile or traditional approach is used, and why.

2.9.4.9 User involvement

The last factor is user involvement. The extent to which users are involved in the software development process is a team factor, as the software developers, business analysts and system analysts have to speak to users to understand their requirements (Hastie & Wojewoda, 2015). It will also be interesting to see if users are involved during all phases of the software development lifecycle, and whether users test software and provide feedback.

2.10 Conclusion

The literature discussed above provides an understanding of software development, agile and traditional approaches, project management, what defines project success and failure, the conceptual framework, and software development in financial institutions. The next chapter discusses the research methodology that was used to investigate software project success within a South African bank.

Chapter 3: Methodology

3.1 Introduction

In the previous chapter, the literature pertaining to software development project success, project failure and failure prevention techniques were discussed. This chapter discusses the research methodology, design, data collection methods, sampling and how data was analysed using different statistical methods.

3.2 Research design

In order to conduct an in-depth investigation of how software development processes work within the digital banking business unit, a case study research design was adopted (Bhattacherjee, 2012). A case study is an empirical inquiry as it investigates the phenomenon within a real life context (Larsson & Ekdahl, 2005). To ensure that validity is achieved in research, case studies allow researchers to conduct a study that is unbiased (Larsson & Ekdahl, 2005). The main goal of any software engineering investigation is to produce results that add value to the body of knowledge (Larsson & Ekdahl, 2005; Mäntylä et al., 2017). Case studies are better suited to software engineering research as they investigate contemporary phenomena that are difficult to study in isolation (Runeson & Höst, 2009). They provide an in-depth understanding of the phenomenon that is being investigated (Runeson & Höst, 2009). A case study that investigates software development project success in a financial institution will therefore contribute to the body of knowledge.

Case studies allow for data to be collected using questionnaires, interviews and personal observations (Bhattacherjee, 2012). They follow an exploratory research approach, offering insights into the area of research (Larsson & Ekdahl, 2005). This research design provides the means to discover a variety of factors, such as social and cultural factors, that may have an impact on the area of research (Bhattacherjee, 2012). A combination of a qualitative and quantitative approach has been adopted, as this allowed for a thorough analysis of the factors that influence the success of software development projects. An interpretive study was adopted in an attempt to understand the participants' understanding of software development project success (Larsson & Ekdahl, 2005). Project managers and software developers provided a diverse sample of team

members from different age groups and backgrounds. This sample also consisted of employees with different levels of experience and skills.

The focus for this study was on the mobile application section of the bank. This section was selected for this case study as each of the units focus on a specific delivery mechanism (online, mobile, cell-phone banking), picking one unit (the mobile application section), attempts to limit the variability of the nature of the development.

3.3 Research approach

Both qualitative and quantitative data allowed for an in-depth investigation into the South African bank chosen, providing important data and answers to many questions on software project success from project managers' and developers' perspectives. Questionnaires and interviews were used to collect the data.

3.4 Study site

The study site was the digital banking business unit within a South African bank, situated in Johannesburg, Gauteng.

3.5 Target population

The target population consisted of 35 software developers and six project managers within the bank's mobile applications digital banking business unit. The researcher is currently employed by this bank as a business analyst in the mobile applications business unit.

3.6 Sampling method

3.6.1 The sampling procedure for quantitative data

Census sampling was used for the questionnaires as the case study's population size was small. The sample size of this study was 35 developers and six project managers as this equalled the total population in each role (41 respondents in total).

3.6.2 The sampling procedure for qualitative data

Extreme case sampling was used as the researcher chose the participants based on certain characteristics. The participants were chosen based on how long they have worked in their roles at this business unit. This ranged from 3 to 15 years (refer to Table 4). The researcher identified two project managers and four software developers for the interviews. No more than six participants were prepared to be interviewed at the time since most of the team members were very busy at the time of data collection (refer to the background section in Chapter 1 for the highly scheduled nature of the development process).

3.7 Data collection methods

This study made use of a combination of a qualitative and quantitative approaches. Questionnaires and interviews were used as the instruments to collect contextual data. Questionnaires were administered to software developers and project managers via Google forms. In-depth interviews were conducted with specific staff, based on experience. The chosen instruments provided the researcher with an effective means to investigate this area of research. The questionnaire can be found in Appendix A: Questionnaire; and the interview schedule can be found in Appendix B.

3.7.1 Permissions needed

A gatekeeper's letter was obtained by the researcher. In order to ensure that participants' rights were protected, the researcher stated their rights in the informed consent in the questionnaire (see Appendix A). All participants agreed to this on the electronic questionnaire before they were allowed to complete any questionnaire questions. Before conducting the interviews, the

researcher also presented each interviewee with a consent form which they had to sign prior to the interview commencing.

3.7.2 Strategies for recording data

An electronic questionnaire was created on Google Forms to collect quantitative data. This was further stored in a Microsoft Excel spreadsheet. Qualitative data was collected by interviewing various project managers and software developers. The researcher asked for consent to record the interview on a cell phone's voice recording app. In addition to recording the interview, the researcher took down key notes from the interview and summarised key points after each interview. The recorded interviews were transcribed in MS Word.

3.7.3 The research instruments used

A questionnaire was used to collect quantitative data and interviews were used to collect qualitative data. A matrix showing how the research questions were operationalised by questionnaire questions and interview questions can be found in Appendix D: Research instrument alignment matrix.

3.7.4 Questionnaires

Questionnaires assist with collecting structured perceptions that can be broken down into numeric data. Electronic questionnaires were used to collect data. Questionnaires are suited to questions that are simple and clear to understand, which require limited responses that can be categorised. This allows for an easier collection of data as a set number of responses will always be provided (Phellas, Bloch & Seale, 2011). Questionnaires are cheap to administer, providing a cost effective solution to researchers, and are not time consuming, allowing participants to complete them in their own time (Phellas et al., 2011). Electronic questionnaires allowed the researcher to collect standard numeric data based on specific perceptions and to investigate the various variables of this study (Bhattacherjee, 2012). Using electronic questionnaires allowed employees to access the questionnaires any time provided they had access to the Internet.

The questionnaire interrogated the factors defined in the theoretical framework (see Figure 1), and other issues identified in the research questions. Project managers and software developers were asked the same questions as this is an overall view of software project success. The questionnaire was made up of three sections. Section A contained demographic questions as well as questions about how long the employee had been working in the role at this business unit. Section B consisted of Likert scale questions related to each factor in the conceptual framework. In an attempt to understand what factors contribute to software development project success, each factor in the conceptual framework was questioned. Each factor had between three and ten questions which respondents used to specify their level of agreement. The final section, Section C, contained two sub sections. Using Likert-scale questions, the respondent had to specify their level of agreement as to whether the stated factors contributed to software development project success or failure. Using a five-point Likert scale, with the options including strongly agree (1), agree (2), neutral (3), disagree (4) and strongly disagree (5), the researcher could categorise the responses in an effective and efficient manner. Where a factor had two or more sub-factors, each sub-factor had a minimum of three Likert scale questions in the questionnaire. This scale measured the constructs indicated by the proposed theoretical framework in an aim to answer the research questions.

3.7.5 Interviews

Interviews are a more flexible form of obtaining information and, when used correctly and efficiently, they can provide researchers with greater insight into, and more context for, the area of research (Phellas et al., 2011). Interviews allow researchers to probe deeper and note down actual reactions to thoughts or ideas; this provides qualitative information which is more valuable than simple surveys (Phellas et al., 2011).

Face to face interviews allow researchers to explain complex questions if necessary. They allow for conversation to flow and thoughts and ideas to be discussed from different perspectives. Interviews allow participants to share freely and with more detail about their experiences, and responses can be clarified immediately. The interviewer has more control and can guide the discussion. Visual aids can also be used to assist with the discussion (Phellas et al., 2011). Body language, voice tone and facial expressions can also give context to the participants' responses (Sekaran & Bougie, 2016).

Interviews were conducted to investigate in more depth how software developers and project managers perceive factors affecting the success of software development projects. The interview questions were developed to target specific employees based on how long they have worked in the business unit. The aim was to identify certain projects that the employees considered successful and unsuccessful and what factors contributed to this success or failure. The interviews provided quality responses and enabled conclusions to be drawn. Verbal cues and body language were very helpful when investigating sensitive information such as job fulfilment and user satisfaction.

The interviews allowed the researcher to obtain answers to open-ended questions. They provided an informative set of responses based on the IT professionals' perceptions and preferences in the bank. Themes were drawn from their responses, thus allowing the researcher to identify patterns in the factors regarding this topic.

3.8 Data collection procedures

3.8.1 Procedures for the collection of quantitative data

Data collection started after ethical clearance was granted (see

Appendix F: Ethical clearance). Due to the fact that the sample size equalled the population, the researcher sent an email containing a link to the electronic questionnaire to all project managers and software developers. The use of Google Forms allowed all questions to be mandatory, and the respondents could not proceed to the next section without completing the previous one. This ensured that all questions were answered in the questionnaire. All respondents were thanked in a follow-up email.

3.8.2 Procedures for the collection of qualitative data

The researcher made a list of which employees to interview. They were chosen based on how long they have worked for the business unit. This ranged from two to six years. An invitation to

a meeting was sent to each of them, and after receiving their responses, the researcher scheduled the interviews.

The researcher had worked with all the interviewees prior to the interview, thus lengthy introductions were not required. The researcher did however, explain the purpose of this study and what she wanted to achieve with the interview. The interviewees were asked to read through the consent forms and were also given a copy of the interview questions prior to the interview. This was done as employees' time is valuable and interviewing them meant taking them away from their jobs. The researcher therefore wanted the interviewees to be prepared in order to ensure that their time was managed effectively. Interviewees had to sign the consent forms before the interviews could start. The interviews were recorded on the researcher's cell phone and notes were taken during the interviews.

3.9 Data collection

Data was collected using the questionnaire and interviews. The questionnaires were distributed on 9 September 2019 and respondents were given a month to complete them. Interviews were conducted once the questionnaires had been returned and the preliminary analysis had been completed. The researcher set up the interviews from 14 October until 16 October 2019.

Google Forms helped the researcher since, as soon as the questionnaires had been completed, the data was recorded in an Excel spreadsheet. The data was then transferred to SPSS in order to start the analysis. This was done as soon as the researcher received the total number of required responses.

The qualitative data from the interviews was recorded and transcribed into Microsoft Word. Under each question, the researcher typed out the responses from the interviewees. This made it easier to understand and to analyse.

3.10 Data Quality Control

Validity and reliability were used to control the quality of the data for this study. These were measured using the data collection instruments. Radio buttons were used on the questionnaires to ensure that the participants did not have to type in their responses. The online questionnaire was pilot tested by one person, who gave feedback about the questions and the structure of the questionnaire as a whole. Since the case study population was small, no respondents were asked to answer the pilot questionnaire, as this would contaminate the findings (Peat et al., 2002). The same questions were asked of each interviewee.

3.10.1 Validity

Validity refers to the ability of data and the research at hand to be logically and physically comprehensive. The validity of the data from this research was ensured by drawing conclusions that correspond to the real world (Bhattacherjee, 2012). To ensure that the content of the questionnaire was valid, the researcher made sure that each question was clear and easy to understand; this is face validity. For content validity, the researcher made sure that the data collection instruments covered all aspects of the conceptual framework. Each question was checked by the supervisor and a statistician, who is an expert in formulating questionnaire questions. Since the questionnaire items were developed for this study, construct validity was tested by using factor analysis. This was how the questionnaire item validity was also determined in a recent study of software development project success (Khoza and Marnewick, 2020). For all factors the percentage of variance extracted, KMO and Reliability (Cronbach's alpha) have been calculated and reported on in Chapter 4. The format on Google Forms was manipulated to ensure that the respondents could easily read the questions.

3.10.2 Reliability

Reliability can be referred to as the degree to which the result of a measurement is accurate or consistent. If the data is dependable it is said to be reliable (Bhattacherjee, 2012). Data is reliable if it produces consistent results when the study is conducted again using the same research instruments. Internal consistency is a measure of the reliability of multiple items under the same construct. To determine this on the questions that made use of the five-point Likert scale, Cronbach's alpha was used.

3.11 Data Analysis

3.11.1 Quantitative data

Quantitative data was drawn from the questionnaires. SPSS was used to analyse this data. The researcher also populated an alignment matrix that showed which questionnaire question answered each research question (see Appendix D: Research instrument alignment matrix). Descriptive and inferential statistical analysis was used to draw conclusions from the questionnaire.

3.11.1.1 Descriptive statistics

Univariate analysis as well as distribution frequency analysis, was conducted on the biographical data. Central tendency analysis provided an understanding of the means, modes and medians of the responses to each factor. Standard deviations also determined how close results were to the mean. Graphs and tables were used to present the data.

3.11.1.2 Inferential statistics

Inferential analysis, correlations and regression tests were used to identify patterns and relationships between the factors. Some of the other statistical tests that were conducted include the:

- Chi-square of goodness-of-fit test to determine if some categories were selected more frequently than others;
- Wilcoxon signed ranked test to determine if the response was different from a neutral Likert scale value of three;
- Pearson and Spearman correlations to determine if there was an association between a factor and a response to other questions;
- Kruskal Wallis and Mann Whitney U tests to compare the perceptions between project managers and software developers for different factors;
- Friedman's test to determine if the average agreement scores for a factor across each category (individual, team, organisational, project) differed significantly.

The themes allowed the researcher to categorise responses creating new factors or variables that the study produced and investigated.

3.11.2 Qualitative data

Qualitative data allows for exploratory and observational results to be uncovered. nVivo was used to analyse the qualitative data. Each question on the interview was structured so that the answer related to a factor in the conceptual framework (see alignment matrix in Appendix D). Based on the data received from the interviews, themes were deduced which were used to relate them to the factors in the conceptual framework.

3.12 Ethical Considerations

A gatekeeper's letter was received from the company being researched. Ethical approval of this research was received from the UKZN Humanities and Social Sciences Research Ethics Committee (HSSREC 02/2019; see

Appendix F: Ethical clearance). This research paper made sure that privacy and confidentiality were upheld. Only those individuals who willingly engaged in the research were used, and a consent form was signed by each respondent. Personal details were not used, and confidentiality was maintained.

3.13 Conclusion

This chapter discussed the research methodology that was used to obtain and analyse the data received from the respondents in order to identify the factors that contributed to software development project success, failure and failure prevention within a South African bank. The next chapter presents the findings and results from the data.

Chapter 4: Results

4.1 Introduction

The purpose of this study was to investigate what factors are critical to the success of software projects from project managers' and software developers' perspectives. This study is based on the conceptual framework presented in literature review section (Chapter 2, Section 2.8). Primary data was collected from questionnaires that were administered to six project managers and 35 software developers. This means that 100% of the 41 respondents answered all the questionnaire questions. To support the results from the questionnaire, two project managers and four developers were interviewed.

The objectives of this study were:

- to determine the extent to which individual factors are critical to the success of software development projects from a project manager's and software developer's perspective within a South African bank;
- to determine the extent to which team factors are critical to the success of software development projects from a project manager's and software developer's perspective within a South African bank;
- to determine the extent to which organisational factors are critical to the success of software development projects from a project manager's and software developer's perspective within a South African bank;
- to determine what factors can prevent software development project failure from a project manager's and software developer's perspective within a South African bank.

Kolmogorov-Smirnov was used to test for normality. Since not all the data was normally distributed, non-parametric data analysis was conducted for this study. The following tests were conducted:

- The Chi-square test was conducted on sections in 4.3, 4.4.1.2, 4.4.1.3 and 4.5.1.1. This is a univariate test, used on a categorical variable to test whether any of the response options are selected significantly more or less often that the others. Under the null hypothesis, it is assumed that all responses are equally selected.

- A Wilcoxon signed ranked test was conducted to determine if the average mean response was significantly different from a neutral Likert scale value of three. This test was also applied in sections 4.4.1, 4.5.1, 4.6.1, 4.8 and 4.9. In sections 4.4.2, 4.5.2 and 4.6.2, which cover the composite values of the factors, the same test was used to determine whether the mean value was significantly different from the mean value of three.
- Spearman's correlation tests were performed to see if each factor correlated with responses to other questions.
- Friedman's test was also used to see if any of the factors ranked significantly lower than the other factors.
- Mann Whitney U and Kruskal Wallis tests were conducted on sections 4.4.2, 4.5.2 and 4.6.2 to see if factor results differed based on gender, race and role.

The findings are presented in this chapter. The chapter first presents the demographic data, followed by the results from the questionnaires and interviews.

4.2 Demographic data

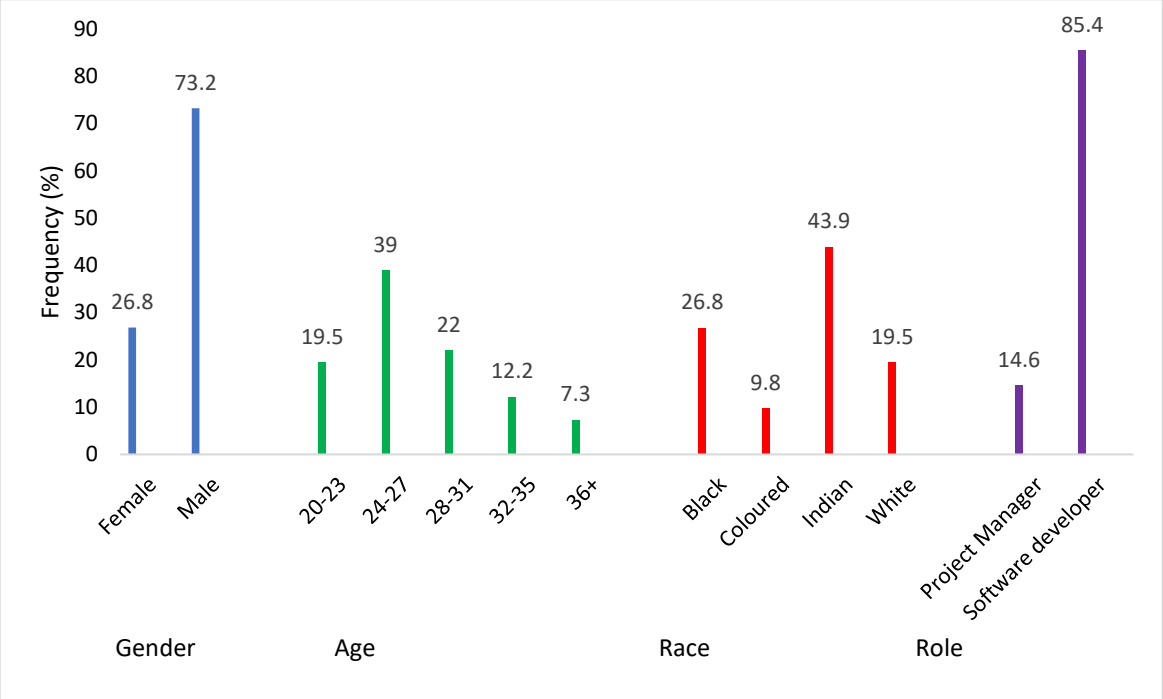

Figure 3: Respondents' demographic information: gender, age, race and role

A total of 41 respondents answered the questionnaire, resulting in a 100% response rate.

Of these, 26.8% were female and 73.2% were male (see Figure 3); and 39.0% were between the ages of 24 and 27. The majority of the respondents were Indian (43.9%), followed by Black (26.8%); with 14.6% of the respondents project managers and 85.4% software developers.

4.3 Employee Roles

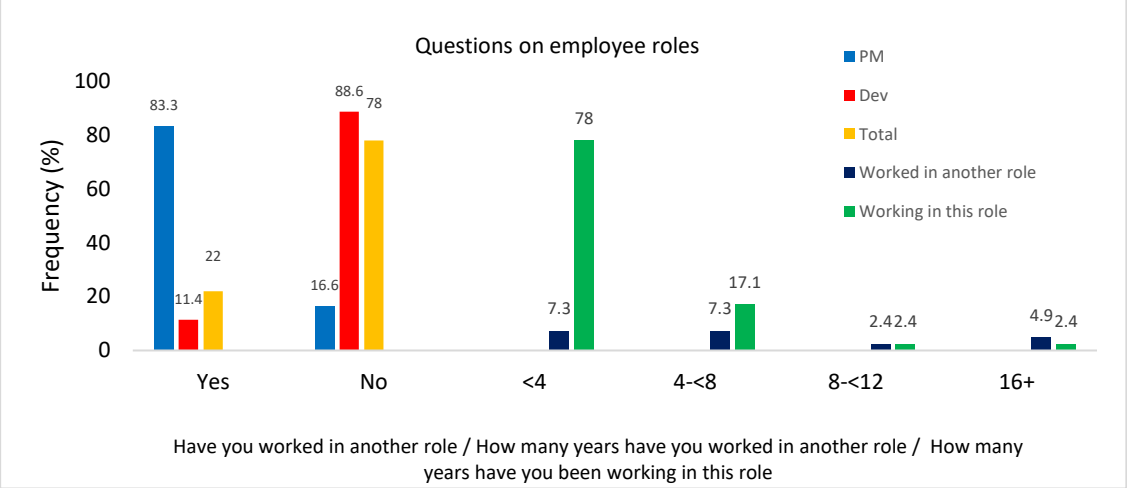

Figure 4: Responses to whether employees had different roles to their current role

Respondents were asked whether they had had different roles compared to their current job roles, and 22% of participants had worked in different roles (see Figure 4). Project managers had worked as client management specialists, business analysts and software developers, and in marketing. Software developers had previously worked as system analysts, full stack developers and IT managers. Figure 4 also shows how many years they had worked in those different roles and how many years employees had been working in their current role in this business unit. A significant 78% indicated that they had worked in the business unit for less than four years X^2 (68.878, 41), $p < .0005$. This indicates that most of the employees, especially software developers, had few years of experience and had only worked in this business unit for most of their careers.

4.4 Number of projects and team members

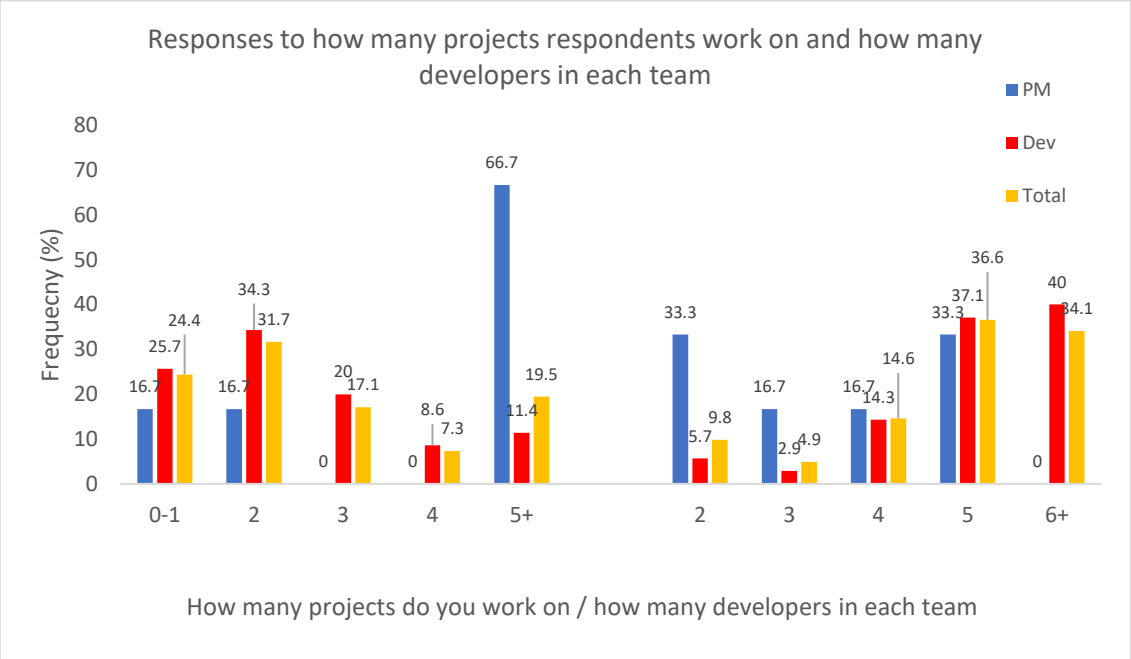

Figure 5: Frequency of responses to how many projects respondents work on and how many developers in each team.

The study asked respondents how many projects they worked on to better understand the organisation and the workload of these respondents. Figure 5 shows the responses to how many projects respondents work on at any given time, as well as how many software developers there are in each software development team. One project manager works on between 0 and 1 projects at a time; one project manager works on two projects at a time; and four project managers work on more than five projects at a time. The project managers that work on no more than two projects at a time have fewer than four years of experience. Developers work on a different number of projects at any given time. Based on Figure 5, it can be seen that most developers work on two projects at a time. Out of the 12 developers (34.3%) who responded that they work on two projects at a time, ten of them had less than four years of experience in this role in this business unit and two had between four and eight years of experience in this role in this business unit. These results offer an insight into this organisation which may relate to other individual, team, organisational and project factors from this study. A bigger development team could result in more communication that is needed and the more effective use of communication methods. The more projects that people work on could relate to the work-life balance factor and job satisfaction.

4.4 Objective 1: Individual factors affecting software project success

The aim of this section was to investigate what individual factors are critical to software project success from project managers' and software developers' perspectives. The objective was to understand whether the level of staff skills, employee experience, work-life balance and job satisfaction are critical for software project success. The results to the questionnaire questions and interviews are first presented (see section 4.4.1), followed by the results from the composite factors (see section 4.4.2). Figure 6 shows the conceptual model used for this study. The individual factors have been highlighted.

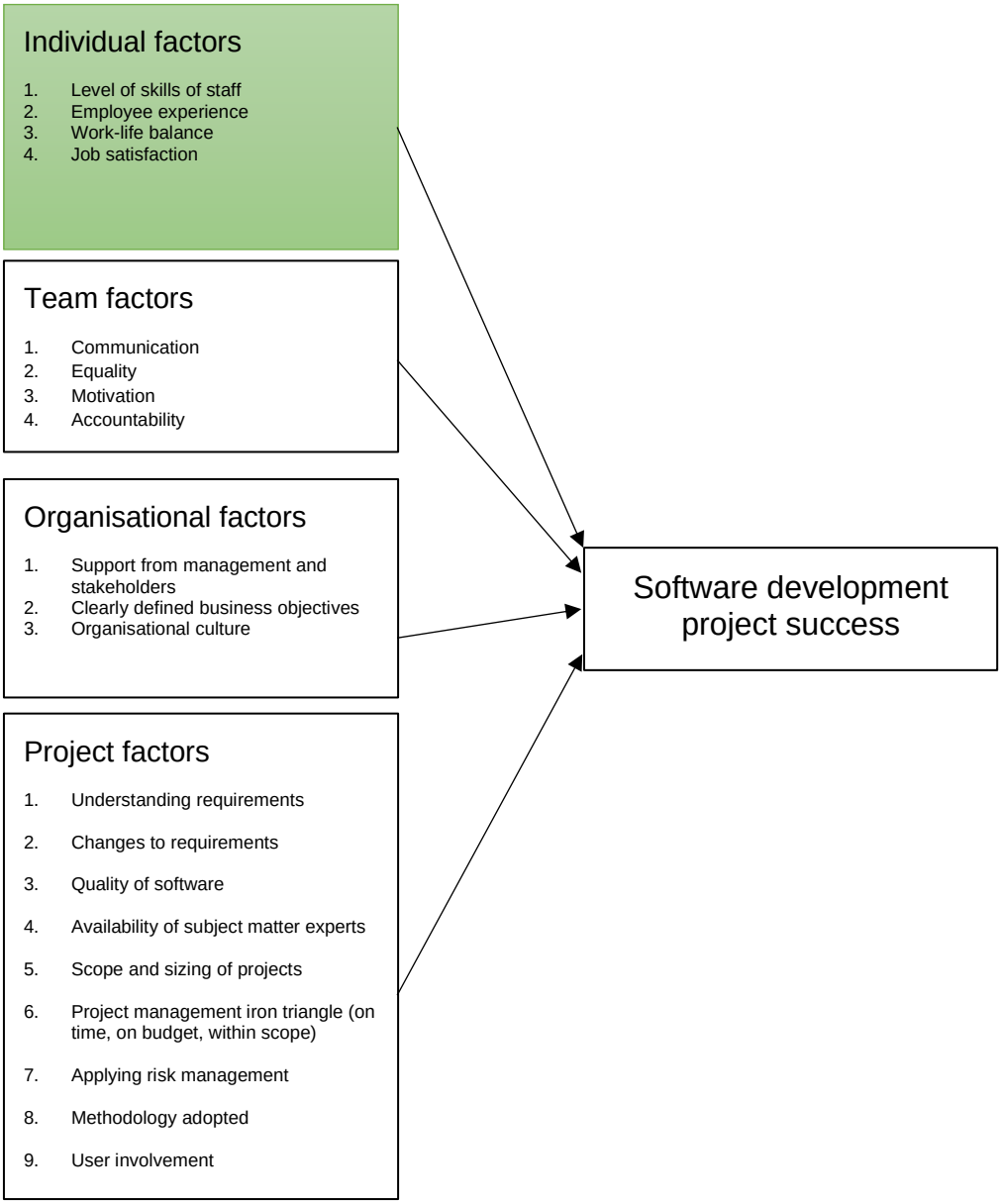

Figure 6: Individual factors in the conceptual framework for the study

4.4.1 Responses to questions about individual factors

This section contains the results of the questionnaire and interview questions for each of the four individual factors.

4.4.1.1 Level of staff skills

In the questionnaire, the level staff skills section (LSS1 to LSS4) asked questions about whether staff had appropriate skills for the job, whether ongoing training is needed for the job, whether solving problems is critical to project success, and whether knowledge in coding and the software development life cycle is critical to success (see Appendix C for the questionnaire). Figure 7 shows that most of the respondents (between 53.7% and 58.6%) strongly agreed that the level of staff skills is critical to project success. Figure 7 also shows how the perceptions of project managers and software developers differ. It is interesting to note that project managers were unanimous in their strong agreement that employees with appropriate skills for the job are critical to success; as well as that ongoing/available training in skills needed for the job is critical to success.

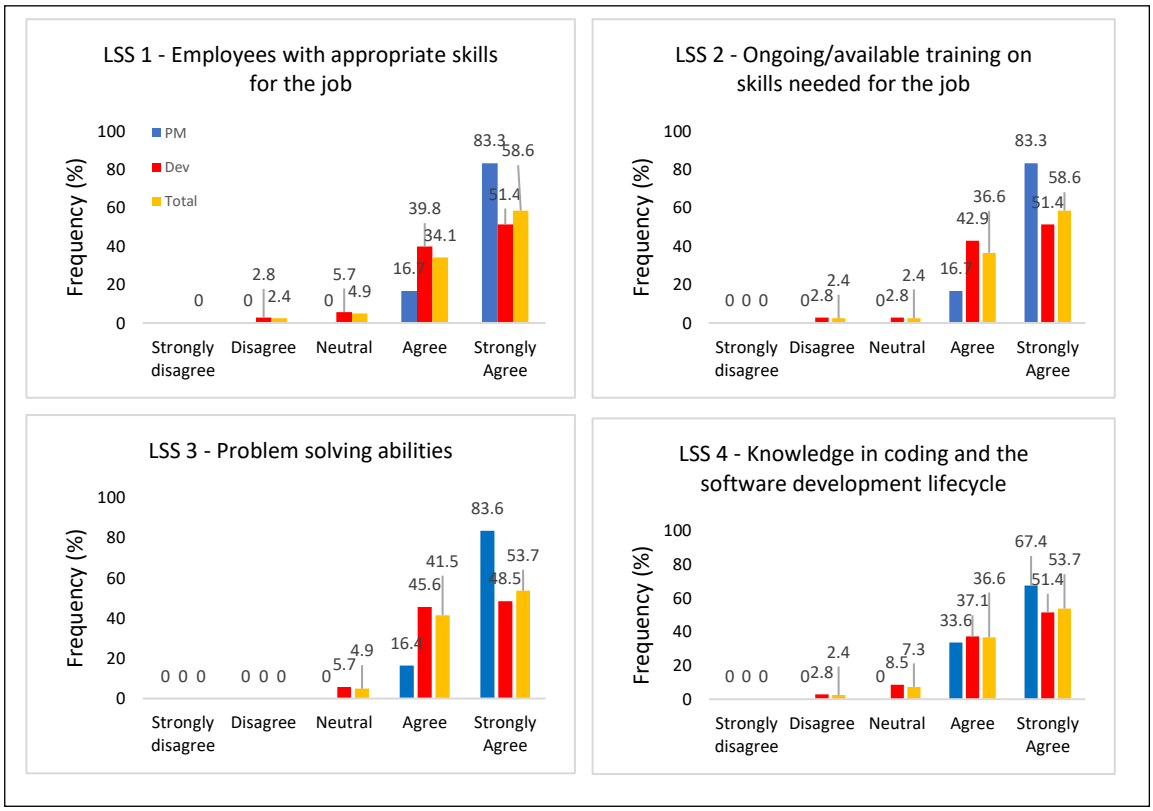

Figure 7: Frequency of responses to whether the level of staff skills is critical to project success

The Wilcoxon signed ranks test is a non-parametric test that determines if the average rank of the responses is significantly different from the central point of the Likert scale (i.e. 3). There is significant agreement to all questions in the level staff skills section. Employees with appropriate skills for the job are critical to project success (Z= -5.527, p<.0005). Ongoing/available training in skills needed for the job is critical to project success (Z= -5.588, p<.0005). Problem solving abilities are critical to project success (Z= -5.622, p<.0005). Knowledge in coding and the software development lifecycle is critical to project success (Z= -5.428, p<.0005).

Employees were interviewed to further understand what factors contribute to project success. When asked if they had the skills necessary to do their jobs, 67% of participants said that they had the necessary skills; but could improve on other aspects, such as communication skills and technical knowledge.

The other 33% of participants disagreed. These participants mentioned that they would like to obtain the necessary skills for their jobs through training. It is worth noting that project managers felt as though they had the necessary skills for their job, while developers were the ones who felt as though they did not have the necessary skills. This seemed to differ from the responses in the questionnaire as, when asked whether appropriate skills are needed for the job, project managers believed that they are critical to project success, whilst developers were also in agreement, but not to the same extent (see LSS1 and LSS4 in Figure 7).

4.4.1.2 Employee experience

Employees were asked, in their opinions, on number of years of experience project managers and software developers needed for a project to be successful. The results are shown in Figure 8.

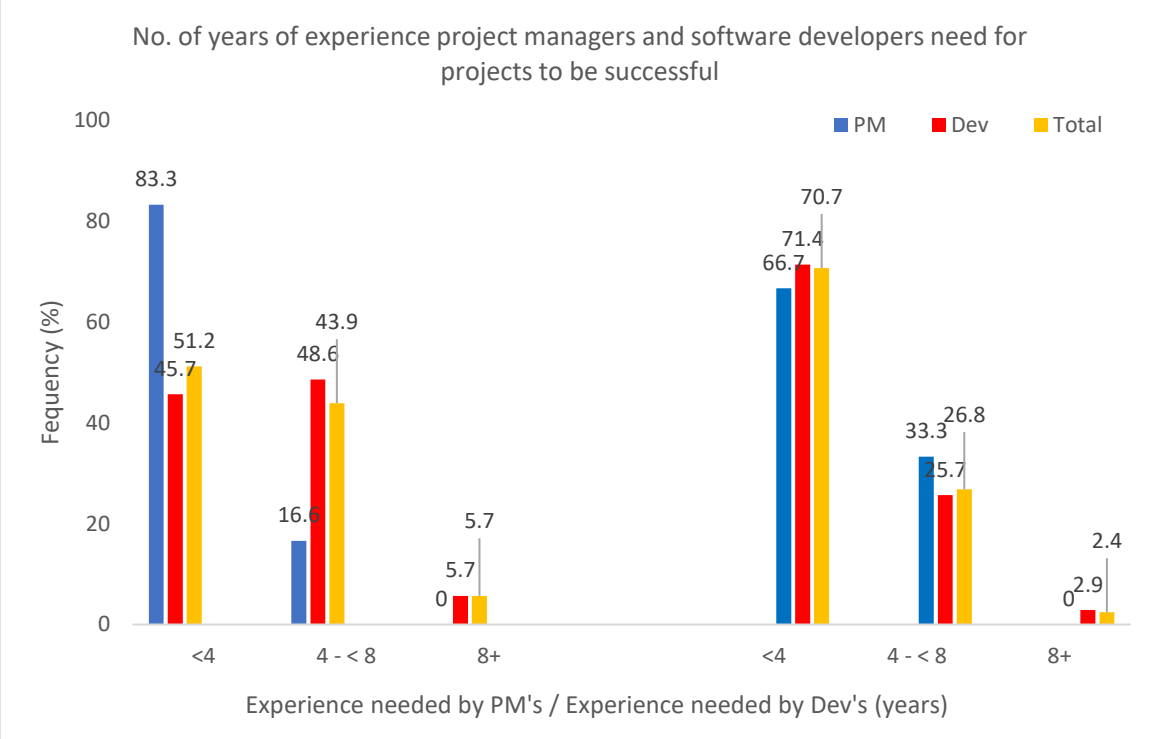

Figure 8: Number of years of experience project managers and software developers need for projects to be successful.

The results in Figure 8 show that project managers believe that having at least four years of experience in project management will be critical to project success. Software developers believe that project managers require more than four years of experience for projects to be successful.

On the right-hand side of Figure 8, 66.7% of project managers believe that software developers having fewer than four years of experience is a contributing factor to success. Software developers are also in agreement that fewer than four years of experience in development ensures that a project will be successful. This group of employees do not necessarily believe that more than four years of experience is vital for project success.

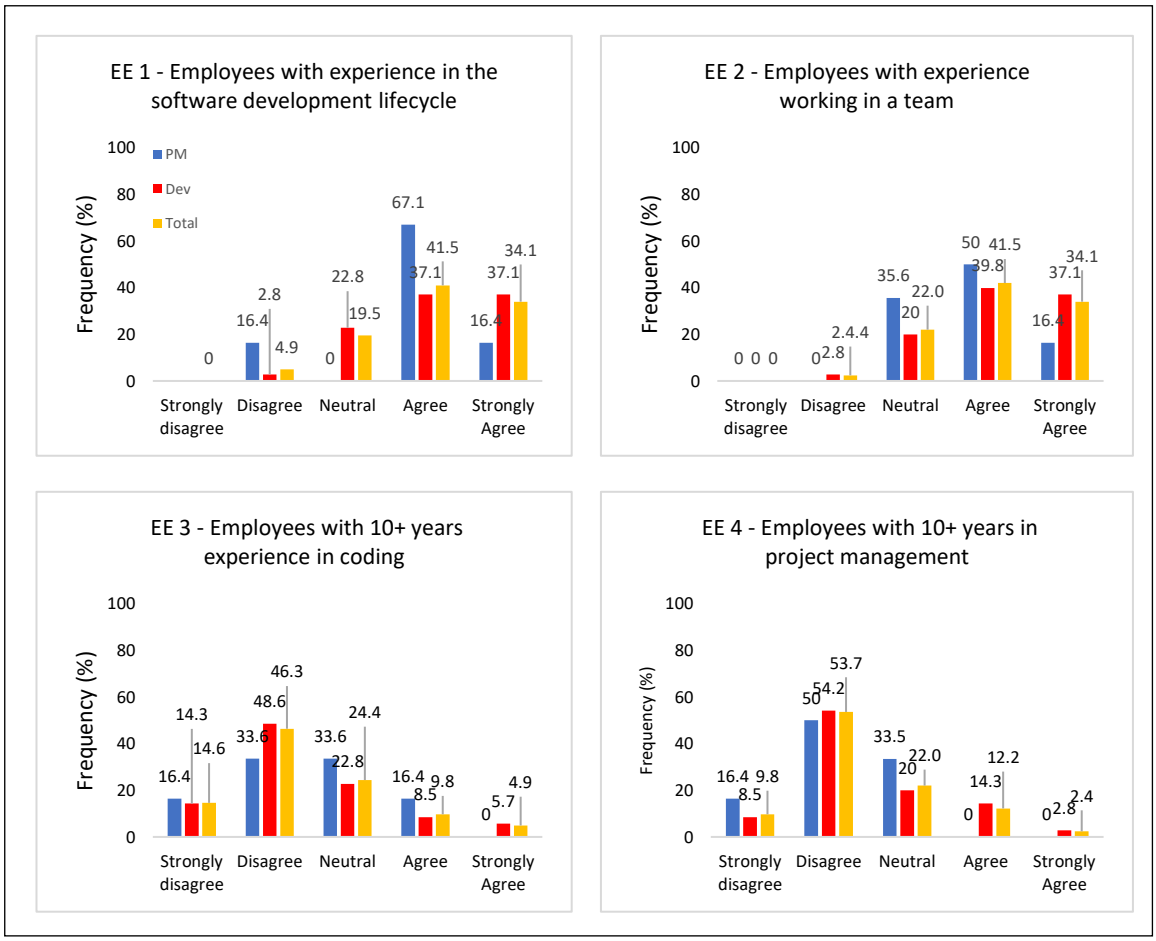

Figure 9: Frequency of responses to whether employee experience is critical to project success

Figure 9 shows the responses from project managers and software developers, based on what employee experience factors they perceive are critical to project success. It can be seen that project managers and software developers hold similar views. The Wilcoxon signed ranks test was conducted to determine if the average rank of the responses is significantly different from the central point of the Likert scale (i.e. 3). Employees with experience in the software development lifecycle are critical to project success ($Z = -4.810$, $p < .0005$). Employees with experience is working in a team are critical to project success ($Z = -4.914$, $p < .0005$). However, there is significant disagreement over whether employees with more than 10 years of experience

in coding are critical to project success (Z= -2.997, p =.003). There is also significant disagreement over whether employees with more than 10 years of experience in project management are critical to project success (Z= -3.315, p =.001).

Based on the results from Figure 8, as well as from EE3 and EE4 (see Figure 9), it can be seen that project managers and software developers in this business unit do not believe that more than 10 years of experience in the respective fields is necessary for a project to be successful. It is also interesting that most of the employees who completed the questionnaire do not have more than four years of experience (see Figure 8).

The employees that were interviewed had between two and six years of experience, as can be seen in Table 2. Only one participant, a project manager, had worked in another software development team before. Of the interviewees, 67% had never worked as a project manager or software developer in another company or business unit before. The remaining 33% had worked in other areas for between eight months and three years. Due to the nature of the results from the questionnaires, interviewees were further questioned on their views whether experience is critical to project success.

Table 2: Participants' experience

PARTICIPANT NUMBER	ROLE	YEARS OF EXPERIENCE IN THIS ROLE IN THIS BUSINESS UNIT	YEARS OF EXPERIENCE IN ANOTHER ROLE IN ANOTHER SOFTWARE DEVELOPMENT TEAM
1	Project manager	3 years	N/A
2	Project manager	5 years	3+ years
3	Software developer	2 years	N/A
4	Software developer	3 years	N/A
5	Software developer	4 years	8 months
6	Software developer	6 years	N/A

Project managers were asked if they needed more than ten years of experience. They disagreed. They believed that project managers all handle projects in their own way and their approaches are different as they consider the team that they are working with and how these team members need to be managed. Project managers, therefore, do not believe more than ten years of experience is necessary for project success. They believed they required between two and four years of experience to fully understand how projects should be handled – within this time they would have worked on all types of projects.

Developers were also asked if they thought project managers needed more than ten years of experience in project management practices. Developers believed that this is not necessary, as experience is not the same as skills. They think a project manager can have the experience but could be lazy, thus not fulfilling their duties; whereas someone new could do the job better, as they may be more motivated or have more technical knowledge that could help improve a project. Developers believed that project managers needed between two and three years of experience.

Project managers agreed that developers do not need more than ten years of coding experience for a project to be successful. They stated that developers who have between two and three years of experience in this development team had obtained all the necessary experience in coding. Project managers believe that, by three years, developers would have learned the necessary development practices to take on any project of any scope and size.

Developers were also asked if they believed that they needed more than ten years of experience for a project to be successful. They all strongly disagreed with this, as they all had less than ten years of experience and felt as though they had the necessary skills to do any project.

This group of employees believe that employee experience is important when it comes to teamwork and knowledge in the software development lifecycle. However, according to these respondents, more than ten years of experience in a field or role is not necessary for a project to be successful. It should be noted that 78% of respondents had less than four years of experience working in their role.

4.4.1.3 Work-life balance

Respondents were asked, on average, how many hours a week they work and how many weekends they work in a month. The results are shown in Figure 10.

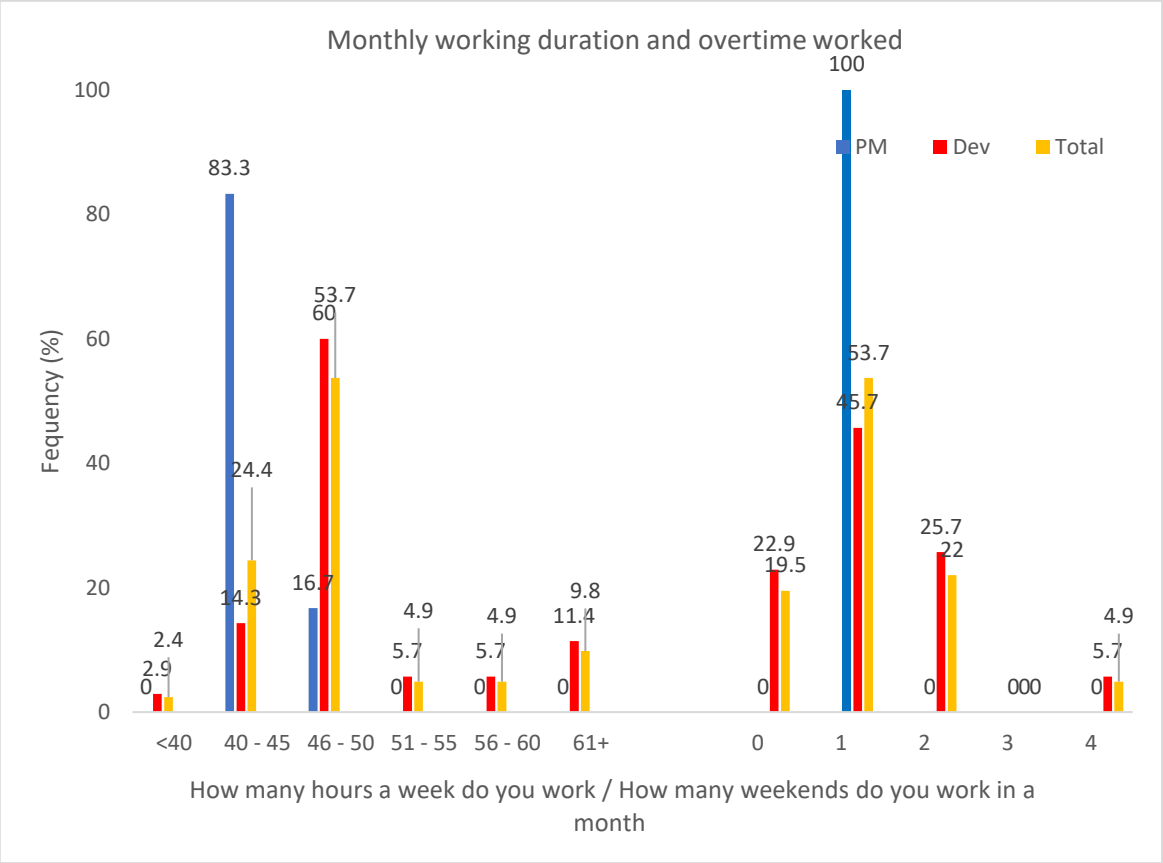

Figure 10: Monthly working duration and overtime worked

The results in Figure 10 show that no project manager works more than 50 hours a week and no project manager works more than one weekend a month. Software developers work more overtime than project managers. It is evident that developers also work on more weekends than project managers. A Chi-square goodness of fit test was conducted to see if some responses were selected more than others. A significant 53.7% indicated that they work between 46 and 50 hours a week, $\mathcal{X}^2$ = (48.122, 41), p<.0005. A significant 53.7% also indicated that they work one weekend a month, $\mathcal{X}^2$ = (20.756, 41), p<.0005.

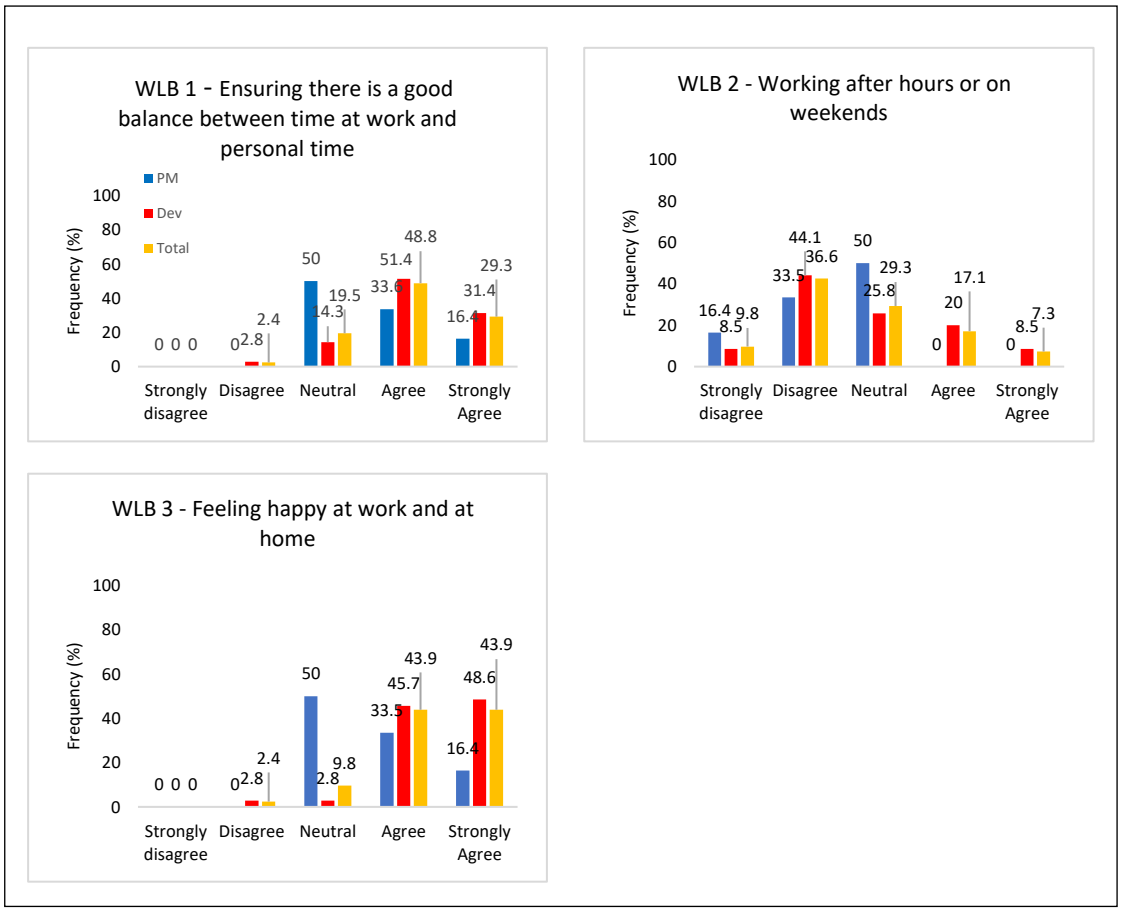

Figure 11: Frequency of responses to whether work-life balance is critical to project success

When respondents were asked whether ensuring there is a good balance between time at work and at home is critical to project success, most agreed, as can be seen in Figure 11. The Wilcoxon signed ranks test was conducted to determine if the average rank of the responses is significantly different from the central point of the Likert scale (i.e. 3). Ensuring there is a good balance between time at work and personal time is critical to project success ($Z = -5.001$, $p < .0005$). It is also interesting to note that, whilst working after hours or on weekends is not seen as a significant factor contributing to success, some software developers still strongly believe that it does ($Z = -1.350$, $p = .177$). The results shown in Figure 10 validate this belief of software developers, as most of the developers work longer hours and, on more weekends, than project managers. Feeling happy at home and at work is critical to software success ($Z = -5.0314$, $p < .0005$). Project managers had conflicting opinions on whether feeling happy at home and at work is critical to project success as 50% were neutral on the topic.

The means and standard deviations were deduced to show how the respondents answered the questions. Table 3 shows how the perceptions of project managers and developers differ. It can be seen that software developers consider the work-life balance questions more important than do project managers, as the means are higher. It is also interesting to note that WLB2 had the highest standard deviation of the factors investigated thus far.

Table 3: Means and standard deviations for the responses to work-life balance

	Project managers	Software developers
	Mean (SD)	Mean (SD)
WLB1 - Ensuring there is a good balance between work time and personal time	3.7 (0.8)	4.1 (0.8)
WLB2 - Working after hours or on weekends	2.3 (0.8)	2.8 (1.1)
WLB3 - Feeling happy at work and at home	3.7 (0.8)	4.4 (0.7)

Table 4 shows that 83% of the interviewees work overtime, ranging from two hours to 12 hours a week. Project managers do not think it is company culture to work overtime. They do not believe that working overtime is a factor contributing to success. This aligns with the results from the questionnaire (see WLB2 in Figure 11). Software developers' answers differed, as some believed that it is company culture to work overtime and some did not. It is interesting to note that some of those who do not think that it is company culture to work overtime still work between two to eight hours of overtime per week.

Table 4: Results from interview questions about overtime.

PARTICIPANT NUMBER	ROLE	HOW MUCH OVERTIME DO YOU WORK?	DO YOU THINK IT IS COMPANY CULTURE TO WORK OVERTIME?
1	Project manager	2 - 4 hours at home and work weekly	No
2	Project manager	3 - 4 hours at home weekly	No
3	Software developer	0	No
4	Software developer	8 hours once a month	No
5	Software developer	8 - 10 hours weekly	Yes
6	Software developer	12+ hours once a month	Yes

Participants five and six appear to work more overtime than the other interviewees. They also believe that it is company culture to work overtime. When participants five and six were asked why their thoughts differed from those of other developers, their responses included:

> "I think it's because I am a senior dev and also a team leader, so I have to make sure that my guys are doing their work properly by reviewing their code" (Participant 5).

Participant 6 mentioned:

> "I like coding, so I am always looking for better ways of doing things. But I think I'm here more than the other guys because a lot of the younger devs come to me with queries, so I need to help them out. I also work on a lot harder stuff than normal screen flows, so have to pay a lot of attention to detail".

4.4.1.4 Job satisfaction

Job satisfaction was investigated through five variables. Respondents answered with either strongly agree, agree or neutral.

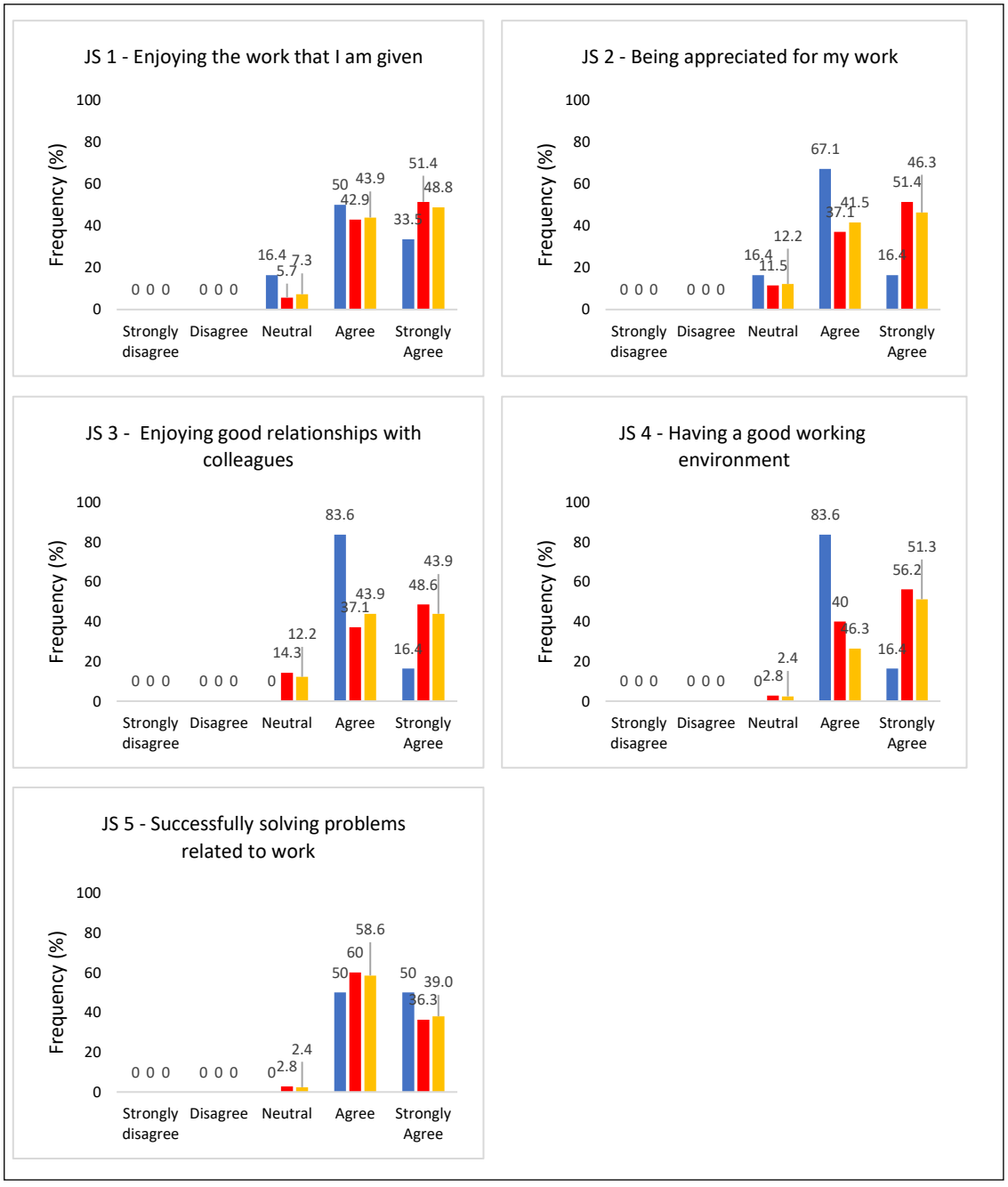

Figure 12: Frequency of responses to whether job satisfaction is critical to project success

Figure 12 shows the results from the questionnaires and also indicates how project managers' and software developers' responses differ. When the Wilcoxon signed rank test was conducted, it

was found that there was no significant disagreement with any of the variables for the job satisfaction questions. 'Enjoying the work I am given' is critical to project success (Z= -5.543, p<.0005). 'Being appreciated for my work' is critical to project success (Z= -5.397, p<.0005). 'Enjoying good relationships with employees' is critical to project success (Z= -5.395, p<.0005). 'Having a good working environment' is critical to project success (Z= -5.686, p<.0005). 'Successfully solving problems related to work' is critical to project success (Z= -5.706, p<.0005).

When employees were interviewed and asked if they enjoy what they do, all interviewees responded that they enjoyed their work (see Table 6). Project managers enjoyed working with people, bringing people together and learning new things every day. Software developers enjoyed solving problems, working on "cool projects", working with different people and learning new skills. This corresponds to the results from the questionnaires where respondents were asked if they enjoy the work they do (see JS1 Figure 12), as well as solving problems (see JS5 Figure 12), were critical to project success. Table 6 shows that project managers are not rewarded for working overtime, whereas developers are rewarded for working overtime. When asked how employees are rewarded for overtime, they all are rewarded with extra money.

Table 5: Results from interview questions about job satisfaction

PARTICIPANT NUMBER	**ROLE**	**ARE YOU REWARDED FOR WORKING OVERTIME?**	**DO YOU ENJOY THE WORK THAT YOU DO?**
1	Project manager	No	Yes
2	Project manager	No	Yes
3	Software developer	Yes	Yes
4	Software developer	Yes	Yes
5	Software developer	Yes	Yes
6	Software developer	Yes	Yes

Table 6 shows that all participants work overtime. However, even though participants 1 and 2 work overtime, they are not rewarded for this. All software developers are rewarded for working overtime. This may be a rule within the company. This may be the reason that developers tend to work more overtime than do project managers (see Figure 10 and WLB 2 in Figure 11).

4.4.2 Composite measures for individual factors

In this section, the composite measures for the four individual factors are presented. Factor analysis was performed to explore the groupings of the items and confirm construct validity. For the LSS and JS factors, one factor was extracted per construct. However, for the EE and WLB constructs, results from a factor analysis were not acceptable as the KMO value was less than 0.6, indicating that the data was not adequate for successful factor extraction to take place. To measure the reliability of each composite construct measure, Cronbach's alpha was calculated. For a reliable measure, alpha should exceed 0.7. When calculating reliability of both the level of staff skills and work-life balance factors, one item needed to be excluded because the item-total correlation was too low. LSS3 (problem solving abilities) was excluded from the level of staff skills composite value, while WLB2 (working after hours or on weekends) was excluded from the work-life balance composite value.

Table 6: Composite measures for individual factors.

Factor	*Items*	*Number of factors extracted*	*Percentage of variance extracted*	*KMO*	*Reliability (Cronbach's alpha)*
Level of skill of staff	LSS1, LSS2, LSS4	1	79.29	.641	.861
Employee experience	EE1 – EE4	*			.731
Work-life balance	WLB1, WLB3	*			.829
Job satisfaction	JS1 – JS5	1	71.00	.844	.896

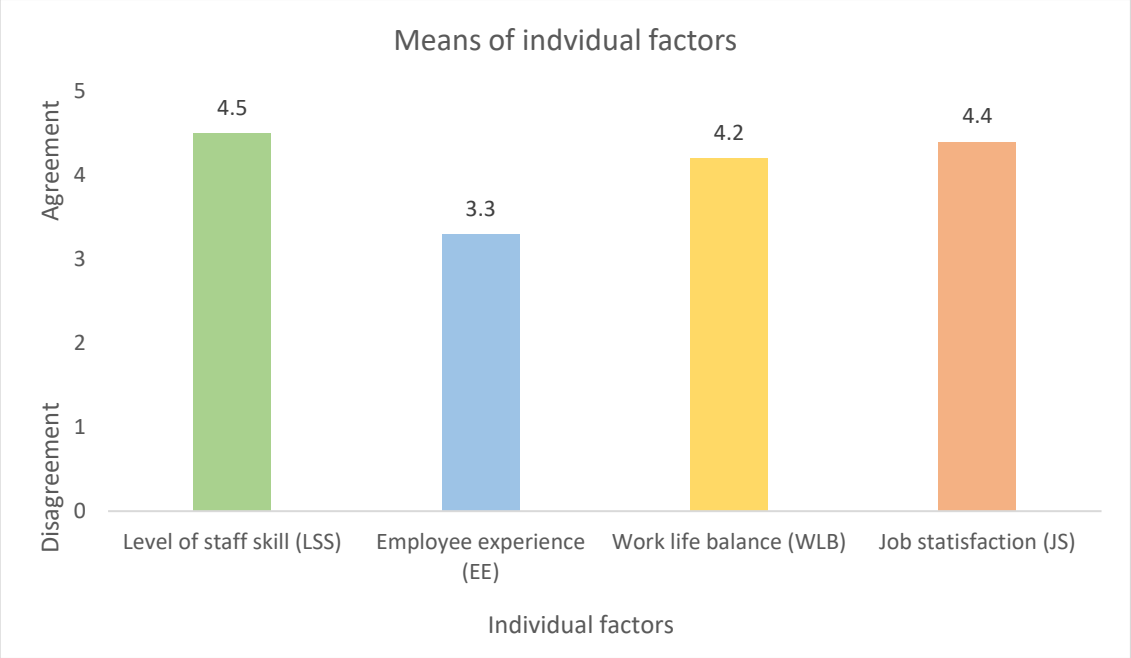

Figure 13: Overall results to responses from project managers and software developers on individual factors

Figure 13 shows the overall values of the resultant means for each individual factor. The Wilcoxon signed rank test was performed on each of the factors if they were significantly different from the mean value of 3. The level of staff skills is a critical success factor ($Z= -5.577$, $p<.0005$). Employee experience is a critical success factor ($Z= -2.174$, $p =.030$). The resultant mean value for employee experience is close to the neutral value of 3. This value may have been different had EE3 and EE4 been rephrased to include a fewer number of years' experience being necessary for project success. Work-life balance is a critical success factor ($Z= -5.244$, $p<.0005$). Job satisfaction is a critical success factor ($Z= -5.601$, $p<.0005$).

A comparison was done to see if the responses differ based on the gender of respondents. No significant difference existed between responses from males or females. Analysis was also conducted to investigate if the responses were different based on race. It was deduced that no significant difference existed in responses based on the race of the respondents. A final comparison was conducted between responses from respondents in the different roles. Software developers consider both work-life balance ($M=4.3$, $p=.003$) and job satisfaction ($M=4.457$, $p=.042$) as significantly more critical to the success of a project than do project managers ($M=3.417$ and 3.967, respectively).

Further analysis (Spearman's correlation test) was conducted to see if there is a correlation between individual factors and responses to questions 1.5 to 1.14. There is a moderately significant correlation between the number of years of experience a project manager should have for a successful project and the agreement that levels of staff skills are critical to success, rho = .413, p=.007. More years of experience is associated with more agreement on the importance of staff skills.

There is also a moderately significant correlation between working overtime and on weekends and employee experience rho = .435, p=.004 and rho = .365, p=.019 respectively. This indicates that employees with more years of experience tend to work more overtime and on weekends. A moderately significant correlation also exists between employee experience and how many years of experience are needed by project managers and software developers for a project to be successful (rho = .393, p=.011 and rho = .496, p=.001 respectively).

A Friedman's test was conducted to see how each of the individual factors ranked. The level of staff skills had a mean rank of 3.15; job satisfaction had a mean rank of 3.01; work-life balance had a mean rank of 2.60; and employee experience had a mean rank of 1.24. Within the individual factors, there are significant differences in perceptions of how critical the four factors are to project success, p<.0005. Specifically, EE is considered significantly less critical than the other three factors.

4.4.3 Conclusion on individual factors

All questions regarding the level of staff skills were shown to be significant, thus making this a critical factor for success. An interesting finding was that project managers either agreed or strongly agreed with statements, whereas some developers responded with disagreement and neutrality.

When respondents were asked about employee experience, two of the four employee experience questions produced results that showed that ten years of experience in a specific role is not critical to project success. This may be because most respondents have less than four years of experience

in this business unit. The composite employee experience (EE) value was considered significantly less critical than the other three factors.

Having a good work-life balance is also a critical factor for success; however, it must be noted that working overtime and on weekends is not critical to project success. Working overtime was also excluded from the composite WLB value. Software developers work more overtime and, on more weekends, than project managers. In addition, developers with more experience tend to work significantly longer hours because they are responsible for junior developers and work on harder projects. That may be the reason why software developers consider having a balance between work and personal time more important than do project managers.

Lastly, ensuring that employees are satisfied in their jobs is critical to project success. It is worth noting that respondents did not disagree with any of the questions for this factor. Once again, software developers' means were higher than project managers', indicating that developers consider job satisfaction more important than do project managers. Employees were also interviewed, and it was noted that project managers are not rewarded for their overtime efforts, whereas developers are. The only two factors in this study where project managers and developers responded significantly differently were for work-life balance and job satisfaction. Developers agreed more strongly with both these factors.

4.5 Objective 2: Team factors affecting software project success

4.5.1 Responses to questions on team factors

The following section contains the results from the questionnaire and interview on team factors. Four team factors were investigated, and project managers and software developers' perceptions were elicited, compared and analysed below. Figure 14 shows the conceptual framework used for this study where the team factors have been highlighted.

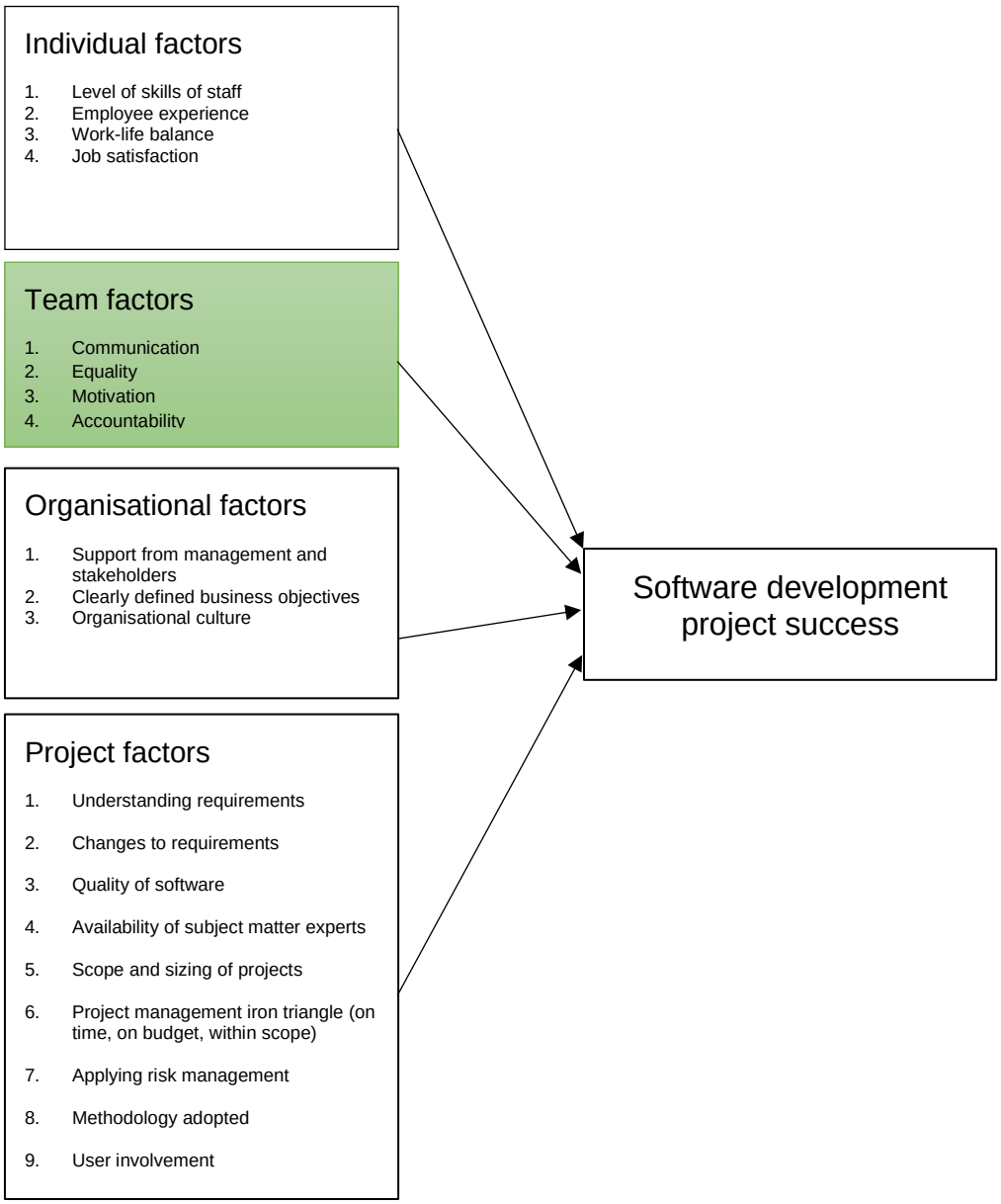

Figure 14: Team factors in the conceptual framework for this study

4.5.1.1 Communication

Communication is the first team factor that was investigated. When asked about communication, respondents mainly responded with 'agree' and 'strongly agree' (see Figure 15). The Wilcoxon signed ranks test was conducted to determine if the average rank of the responses is significantly different from the central point of the Likert scale (i.e. 3). Good communication between team members is critical to project success (Z= -5.674, p<.0005). Project managers were unanimous in their agreement with this question (see Figure 19). Team members knowing about each other's projects is a critical success factor (Z= -5.061, p<.0005). This question did, however, show some disagreement amongst software developers. Communicating in person with team members about projects is critical to project success (Z= -4.487, p<.0005). Project managers believed more that this was necessary to project success than developers did. Regular communication with team members regarding projects is also critical to project success (Z= -5.098, p<.0005). One software developer strongly disagreed that this is necessary for project success.

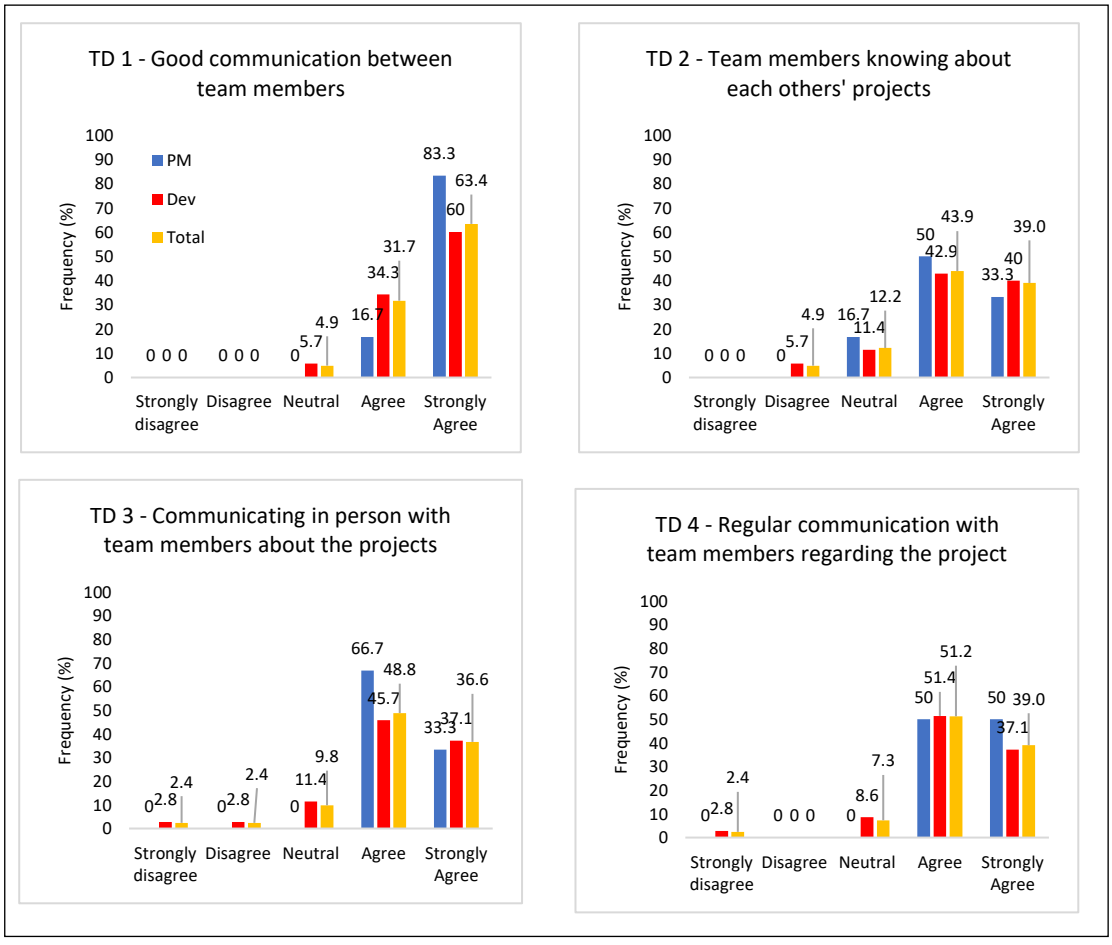

Figure 15: Frequency of responses to whether communication is critical to project success

4.5.1.2 Being treated fairly

Figure 16 shows the responses to whether being treated fairly is critical to project success. The Wilcoxon signed rank test shows that all team members being treated the same is critical to project success (Z= -5.614, p<.0005). Project managers' agreement or strong agreement is unanimous on this question. Two software developers (5.7%) responded with 'neutral'. All team members being given the same amount of work is critical to project success (Z= -4.938, p<.0005). One software developer (2.8%) disagreed with this question. Work being allocated in a fair way to all members in the team is critical to success (Z= -5.401, p<.0005). One software developer (2.8%) disagreed and three (8.6%) responded with neutrality. The developer who disagreed with TD6 and TD7 has been working in this business unit for less than four years and has been working in this role for between four and eight years. It is unclear if this developer has a more senior role.

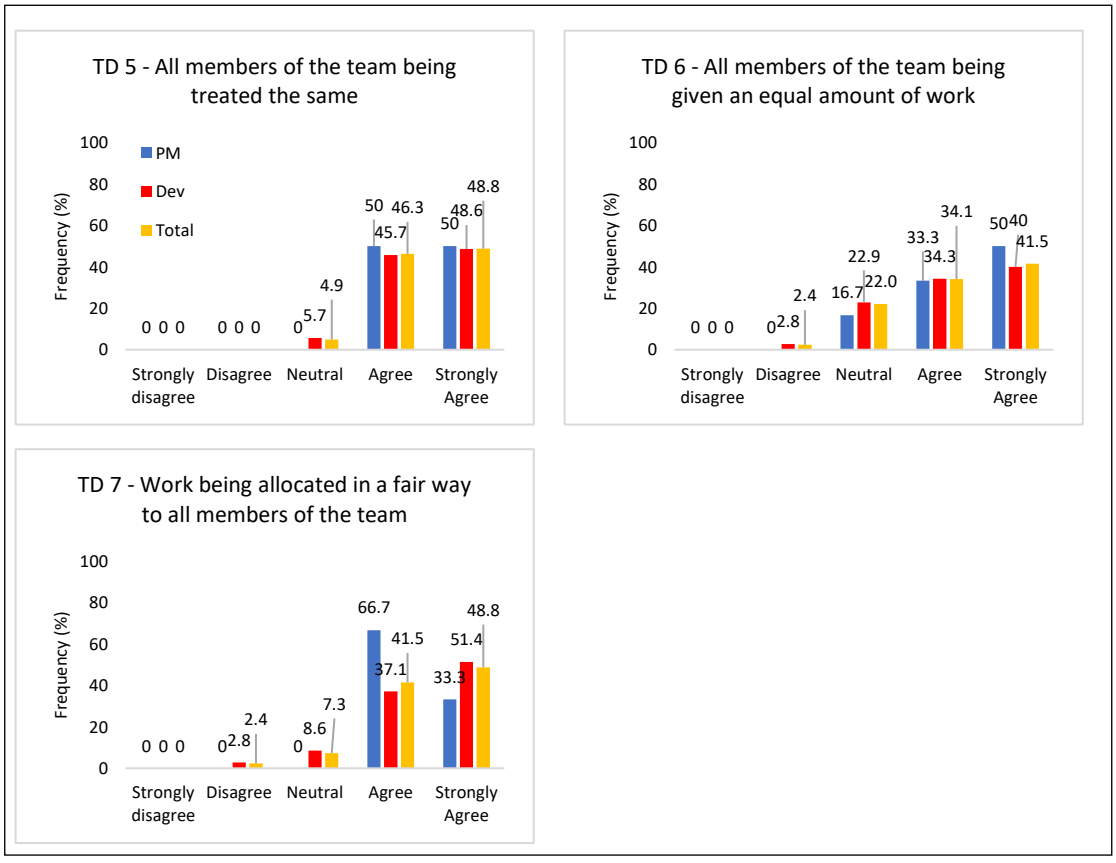

Figure 16: Frequency of responses to whether being treated fairly is critical to project success

4.5.1.3 Accountability

Respondents believe that being accountable is vital to project success. Based on Figure 16, 48.8% of respondents 'agreed' and 'strongly agreed' that all team members taking responsibility for a project is critical (Z= -5.608, p<.0005). One developer (2.8%) seemed to disagree that taking responsibility is critical to project success. All team members being accountable for their aspect of the project is critical to success (Z= -5.684, p<.0005). Again, a different developer disagreed that this is critical for a project to be successful. Managers addressing team members who are not accountable is a critical success factor (Z= -5.617, p<.0005). All project managers and most software developers were unanimous in their agreement that this is critical for projects to be successful. The developer who disagreed with TD8 had less than four years of experience working in this business, but between four and eight years working in this role. The developer that disagreed with TD9 had less than four years of experience overall.

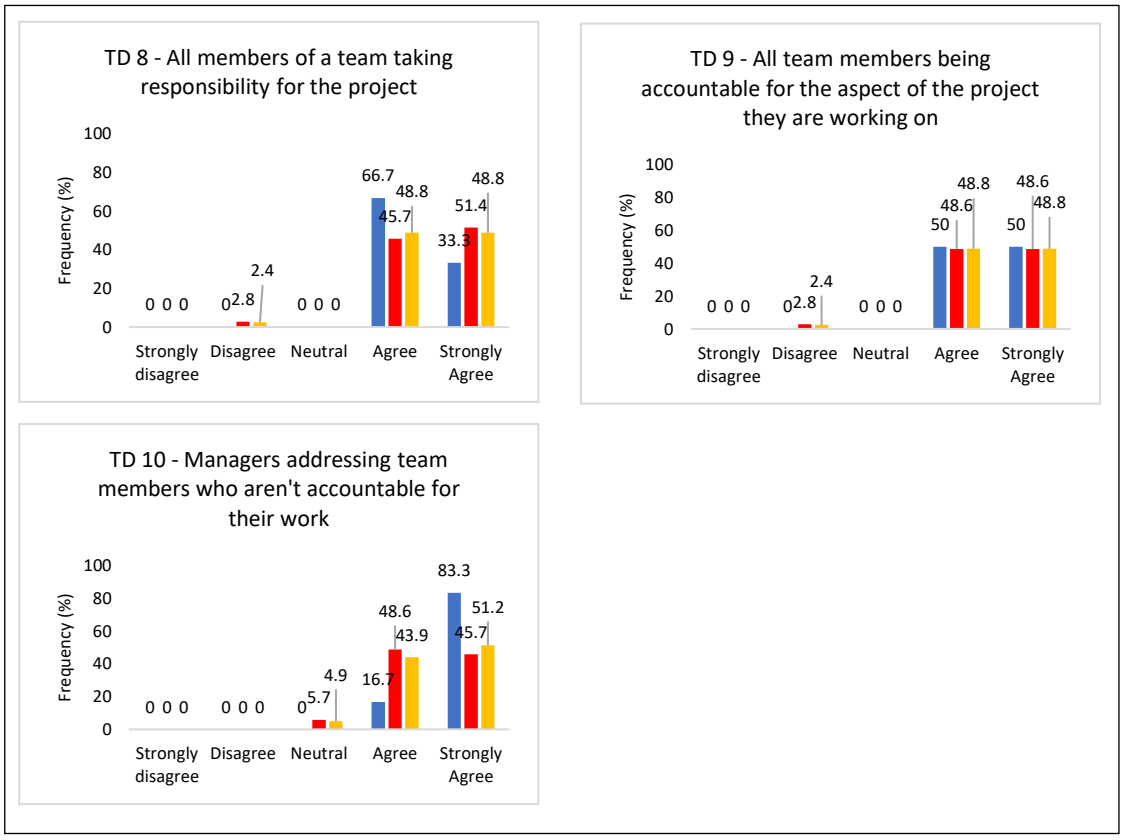

Figure 17: Frequency of responses to whether being accountable is critical to project success.

4.5.1.4 Motivation

Respondents indicated their agreement as to whether team motivation affected project success. Based on the results shown in Figure 18, team motivation is also a success factor.

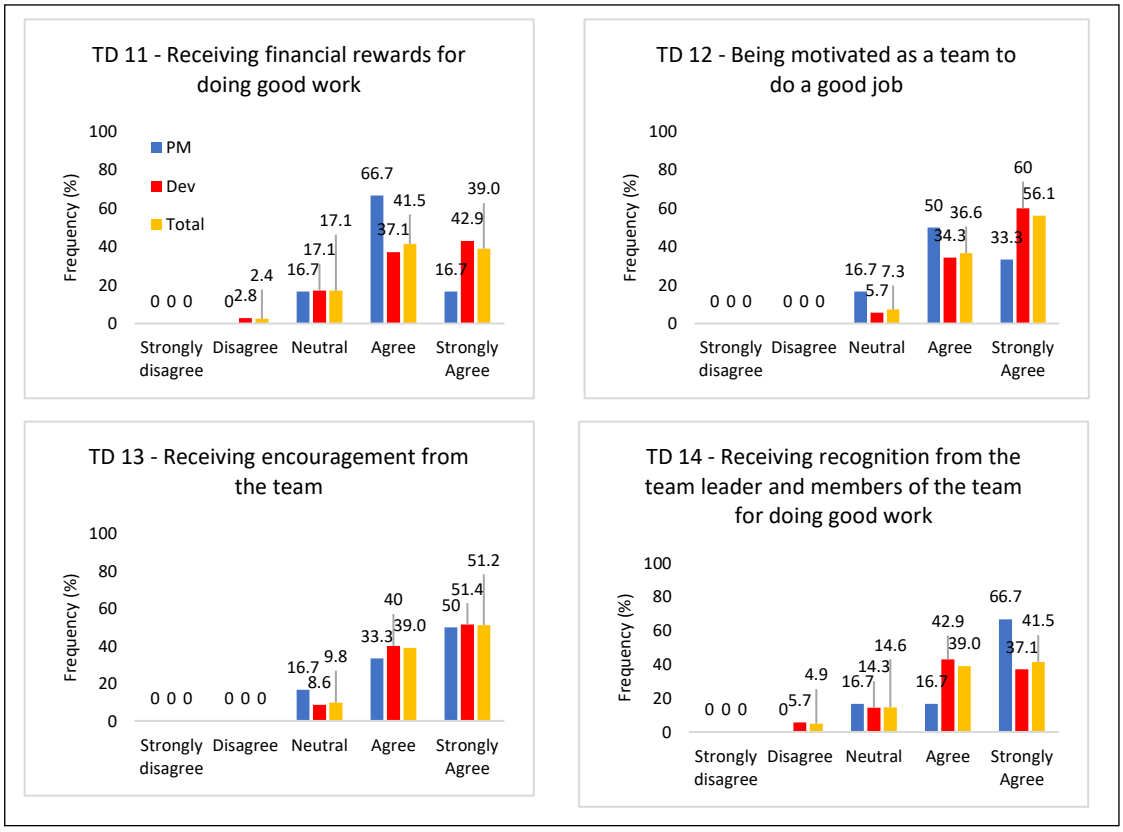

Figure 18: Frequency of responses to whether motivation is critical to project success.

The Wilcoxon signed ranks test was conducted to determine if the average rank of the responses is significantly different from the central point of the Likert scale (i.e. 3). Receiving financial rewards for doing good work is critical to project success ($Z = -5.079$, $p < .0005$). However, one developer disagreed. Being motivated as a team to do a good job is critical to project success ($Z = -5.565$, $p < .0005$), even though some project managers and software developers responded with 'neutral'. Receiving encouragement from the team is critical to project success ($Z = -5.479$, $p < .0005$). Receiving recognition from the team leader and members of the team for doing good work is also critical to project success ($Z = -5.000$, $p < .0005$). Two software developers (5.7%) seem to disagree with this question.

4.5.2 Composite measures

In this section, the composite measures for the all the team factors are presented. In order to measure reliability, the Cronbach's alpha test was performed.

Factor analysis was performed on team sub factors, communication, equality, accountability and motivation. It successfully produced three factors, i.e. communication, being treated fairly and motivation. Accountability was included with the communication factor. Table 7 shows the results from the factor analysis on questions on team dynamics. It can be seen that the two accountability factors (TD8 and TD10) were included in the communication factor. TD9 was excluded from the factor analysis. The factors account for 71.48% of the variance.

Table 7: Factor analysis on team dynamics

	Factor		
	Communication	Being treated fairly	Motivation
TF TD3 Communicating in person with team members about the projects	0.954		
TF TD4 Regular communication with team members regarding the project	0.945		
TF TD8 All members of a team taking responsibility for the project	0.7		
TF TD10 Managers addressing team members who aren't accountable for their work	0.538		
TF TD1 Good communication between team members	0.537		
TF TD2 Team members knowing about each other's projects	0.401		
TF TD12 Being motivated as a team to do a good job		0.818	
TF TD13 Receiving encouragement from the team		0.803	
TF TD11 Receiving financial rewards for doing good work		0.766	
TF TD14 Receiving recognition from the team leader and members of the team for doing good work		0.56	

TF TD6	All members of the team being given an equal amount of work	0.878
TF TD7	Work being allocated in a fair way to all members of the team	0.877
TF TD5	All members of the team being treated the same	0.476

Table 8: The sub factors produced from team dynamics.

	Sub factors	KMO	Reliability (Cronbach's alpha)
Sub factor 1	Communication	.69	.878
Sub factor 2	Being treated fairly	.69	.852
Sub factor 3	Motivation	.69	.802

Table 9: Composite measures for team factors.

Factor	*Items included*	*Items excluded*	*Resultant Cronbach's alpha*	*Z*	*P*
Communication/ Accountability	TD1 – TD4, TD8, TD10	TD9	0.878	-5.570	<.0005
Being treated fairly	TD5 – TD7	-	0.852	-5.539	<.0005
Motivation	TD11 – TD14	-	0.802	-5.516	<.0005

Figure 19 shows the overall resultant means for each team factor. All factors are significant. A comparison was done to investigate if the results differed, based on the gender of participants. No significant difference was found. Analysis was also done to see if results differed based on the race of each participant. Once again, no significant difference was recorded. A final comparison was conducted to see if results differed based on the participants' roles. No significant difference was found.

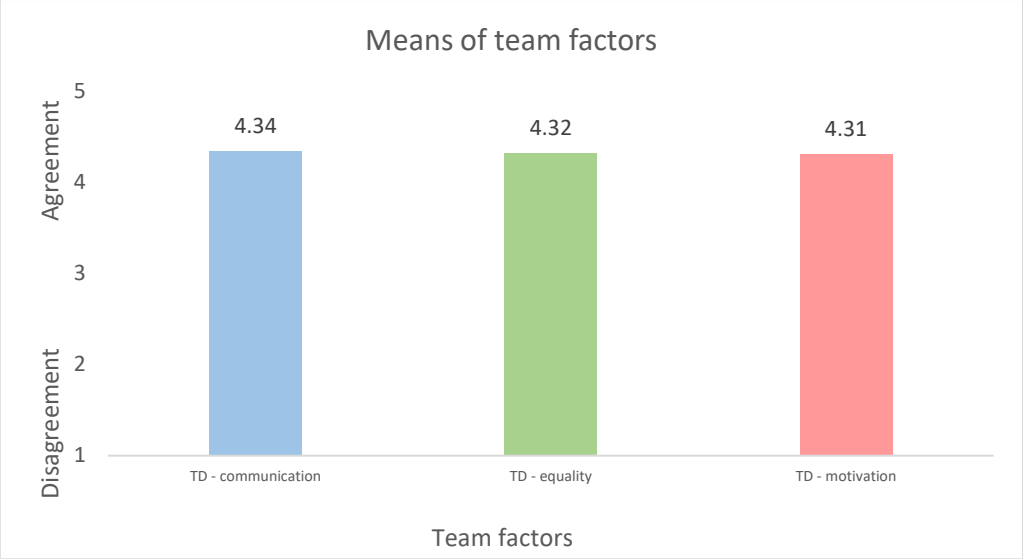

Figure 19: Overall results to responses on team factors.

Further analysis was conducted (Spearman's test) to see if there existed a correlation between team factors and questions 1.5 to 1.14. There is a moderately significant relationship between the team factor, equality, and how many developers are in a project team, rho = .314, p = .045. This means that if there are more developers in a team, it is necessary for each member to be treated equally to ensure project success.

4.5.3 Conclusion on team factors

The team factors were analysed and discussed above. All factors were found to be significant and critical to project success. No one factor was significantly less important than any other. There were also no significant differences between the different genders, races and roles.

4.6 Objective 3: Organisational factors affecting software project success

The aim of this section was to investigate what organisational factors are critical to software project success from project managers' and software developers' perspectives. The objective was to attempt to understand whether support from management and stakeholders, clearly defined business objectives and organisational culture is critical for project success. The results from the questionnaire and interview questions are presented first (see Section 4.6.1), followed by the results from the composite factors (see Section 4.6.2). Figure 20 shows the conceptual framework with the organisational factors highlighted.

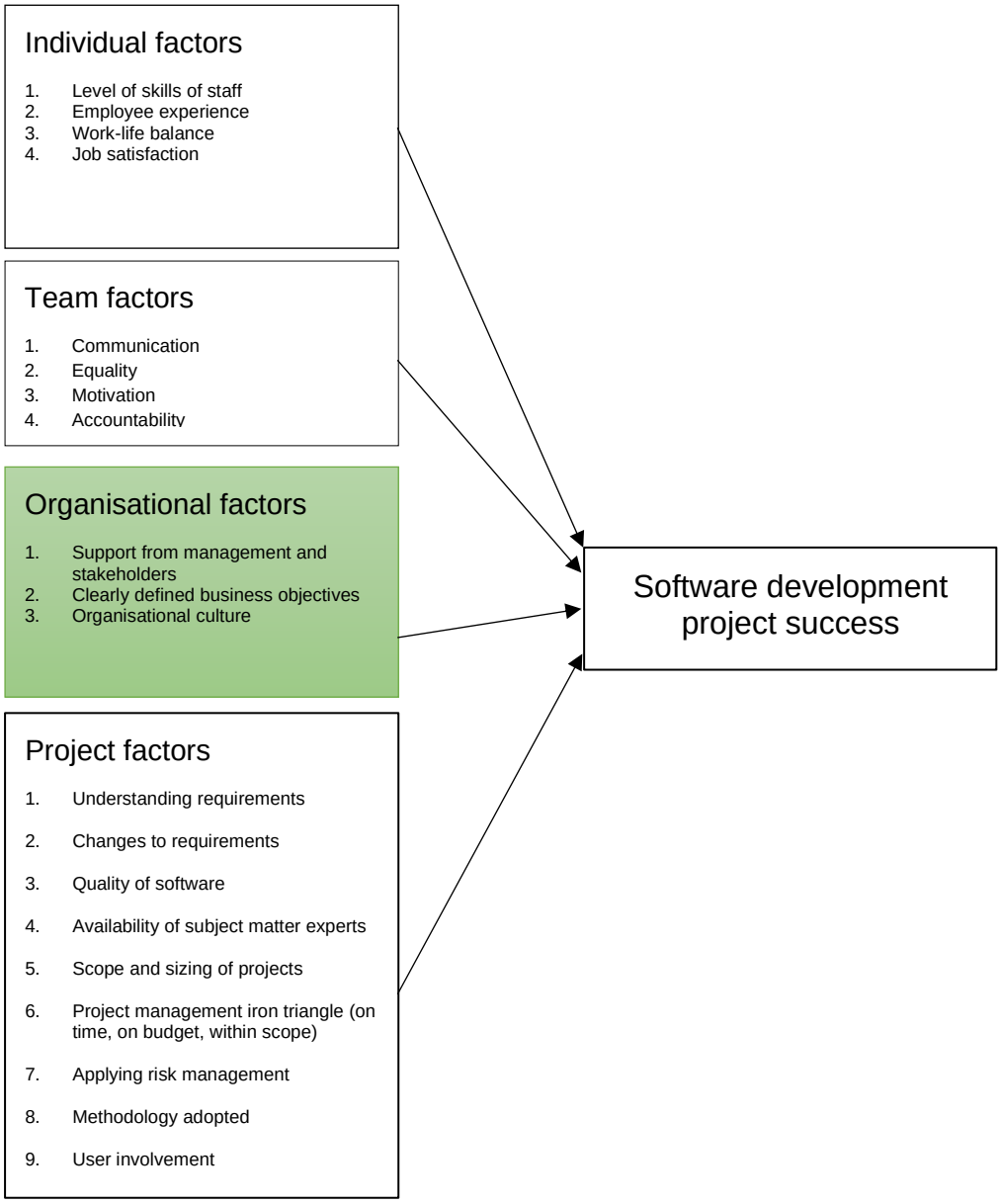

Figure 20: Organisational factors in the conceptual framework.

4.6.1 Responses to questions on organisational factors

The section below reports on the perceptions of three organisational factors as expressed in the questionnaires and interviews.

4.6.1.1 Support from management and stakeholders

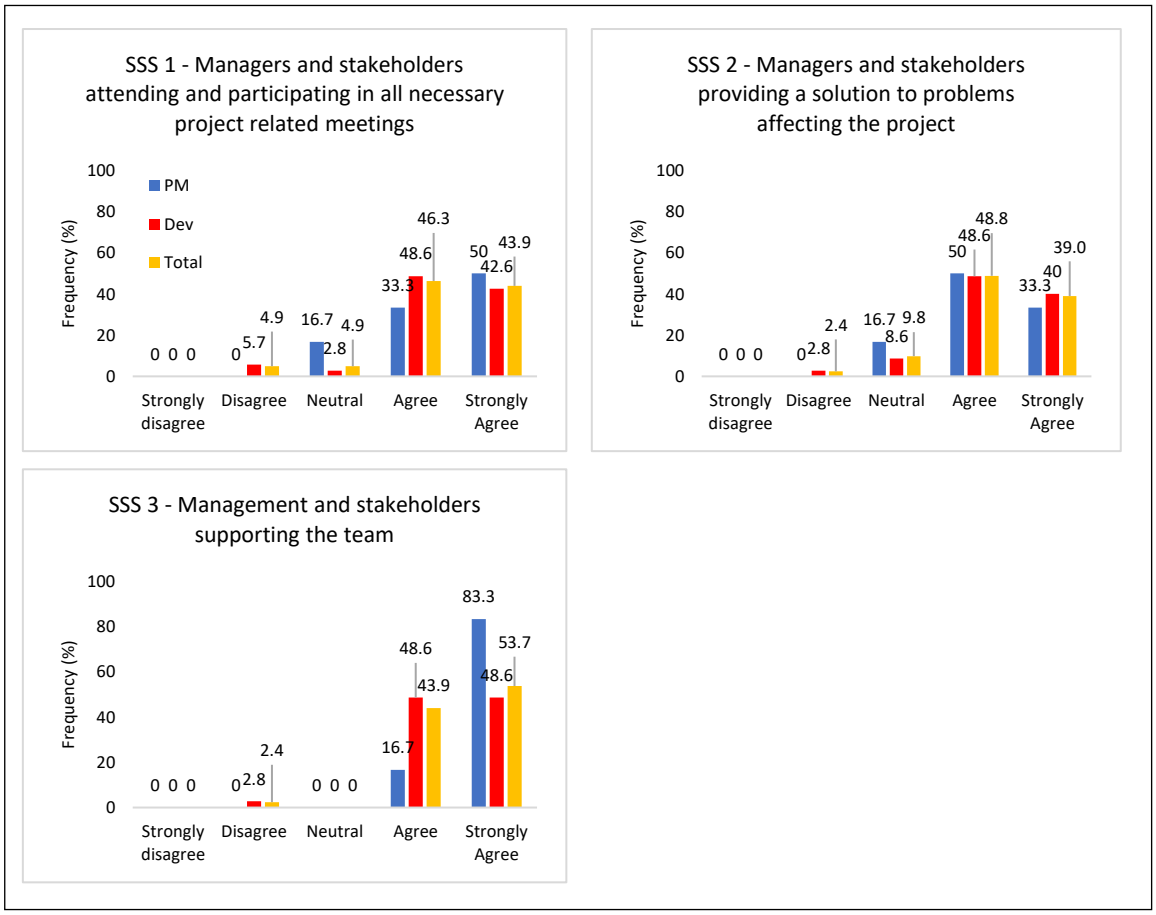

Figure 21: Frequency of responses to whether support from management and stakeholders is critical to project success.

Support from management and stakeholders is a success factor, as indicated in Figure 21. It is important for managers and stakeholders to provide support and participate in the project, as indicated by respondents agreeing and strongly agreeing with the above statements. The Wilcoxon signed ranks test was conducted to determine if the average rank of the responses is significantly different from the central point of the Likert scale (i.e. 3). Managers and stakeholders attending and participating in all necessary project-related meetings is critical to

project success (Z= -5.300, p<.0005). It is also vital that they be involved in problem resolution, shown by 48.8% of respondents agreeing with the statement. Managers and stakeholders providing a solution to problems affecting the project is critical to project success (Z= -5.307, p<.0005). Management and stakeholders supporting the team is critical to project success (Z= -5.623, p<.0005). It is interesting that project managers answered SSS3 in the same way as TD10 – addressing the team who aren't accountable for their work. This seems to be one of the areas in which project managers need support.

When employees were interviewed to understand how they thought management can support the team more, they responded by stating that management needs to be more involved in projects. For example, a project manager responded:

> "Managers can be involved in meetings regarding the project solution design. They are usually the ones who want the projects on the roadmap, but never attend any project-related meetings." (Participant 1).

Management should be transparent and communicate the goal of the project to the team, ensure that everyone in the team does their jobs, and have regular sessions with the team. This can be compared to the result from TD10, where project managers 'agreed' and 'strongly agreed' that managers addressing team members who aren't accountable is critical to success (see Figure 30). While most software developers also agreed, two responded with 'neutral'.

Management also needs to support the team by raising issues earlier so that they are resolved quicker. When interviewees were asked how management can support individuals more, they responded with: assisting when help is needed; being more available to the team; talking to them frequently and providing guidance to team members.

4.6.1.2 Clearly defined business objectives

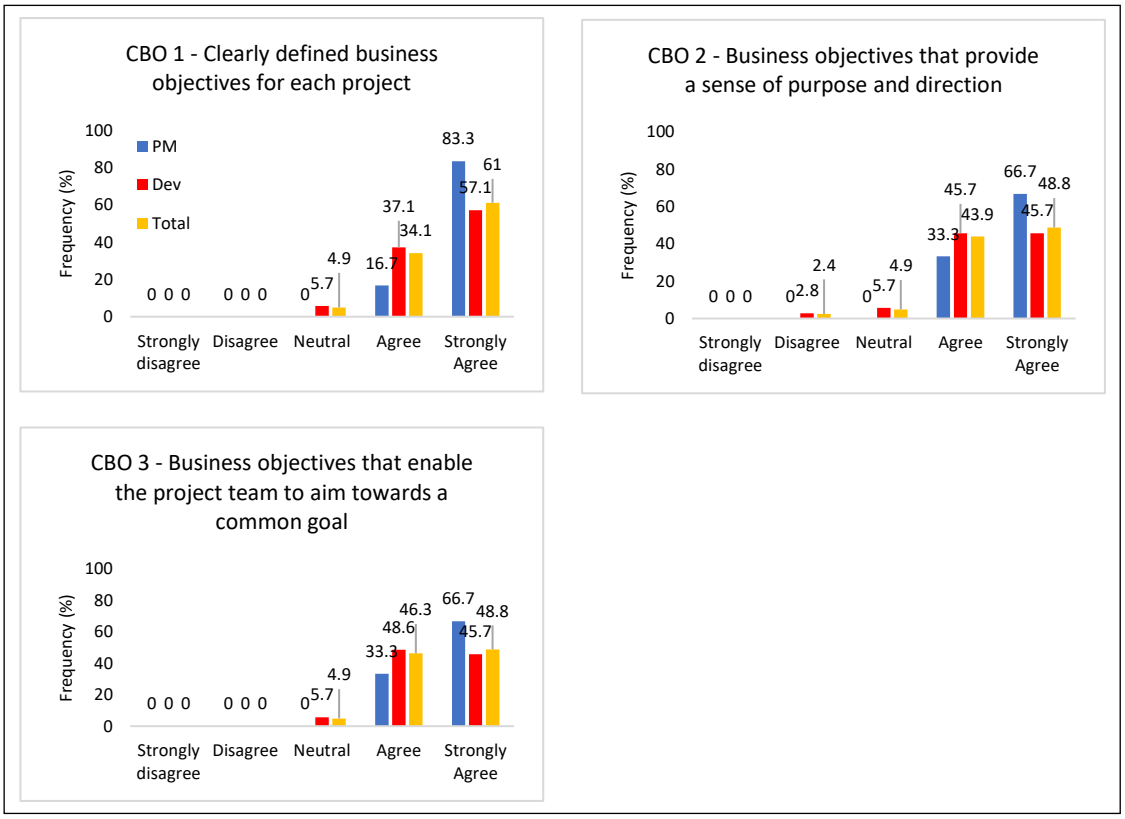

Figure 22: Frequency of responses to whether clearly defined business objectives are critical to project success.

The Wilcoxon signed ranks test was conducted to determine if the average rank of the responses is significantly different from the central point of the Likert scale (i.e. 3). Clearly defined business objectives for each project are critical to project success (Z= -5.657, p<.0005). Of the respondents, 61% strongly agreed that clearly defined business objectives lead to project success, while 48.8% strongly agreed that business objectives provide a sense of purpose and direction and provide the project team with a common goal to work towards. Business objectives that provide a sense of purpose and direction are critical to project success (Z= -5.470, p<.0005). Business objectives that enable the project team to aim towards a common goal are critical to project success (Z= -5.614, p<.0005).

4.6.1.3 Organisational culture

Six questions were asked about organisational culture (see Figure 23). All factors are significantly different from the central point of the Likert scale (i.e. 3), as indicated by the results of the

Wilcoxon signed ranks tests. A friendly working environment is important for project success, as indicated in Figure 27 (Z= -5.631, p<.0005). Respecting each other's work environment is a success factor (Z= -5.155, p<.0005). A supportive working environment is a critical success factor (Z= -5.622, p<.0005). Empathy shown by the organisation in times of personal need is another success factor (Z= -5.631, p<.0005). Leadership setting an example and following company values will ensure project success (Z= -5.160, p<.0005). Following company values is also a success factor (Z= -5.181, p<.0005).

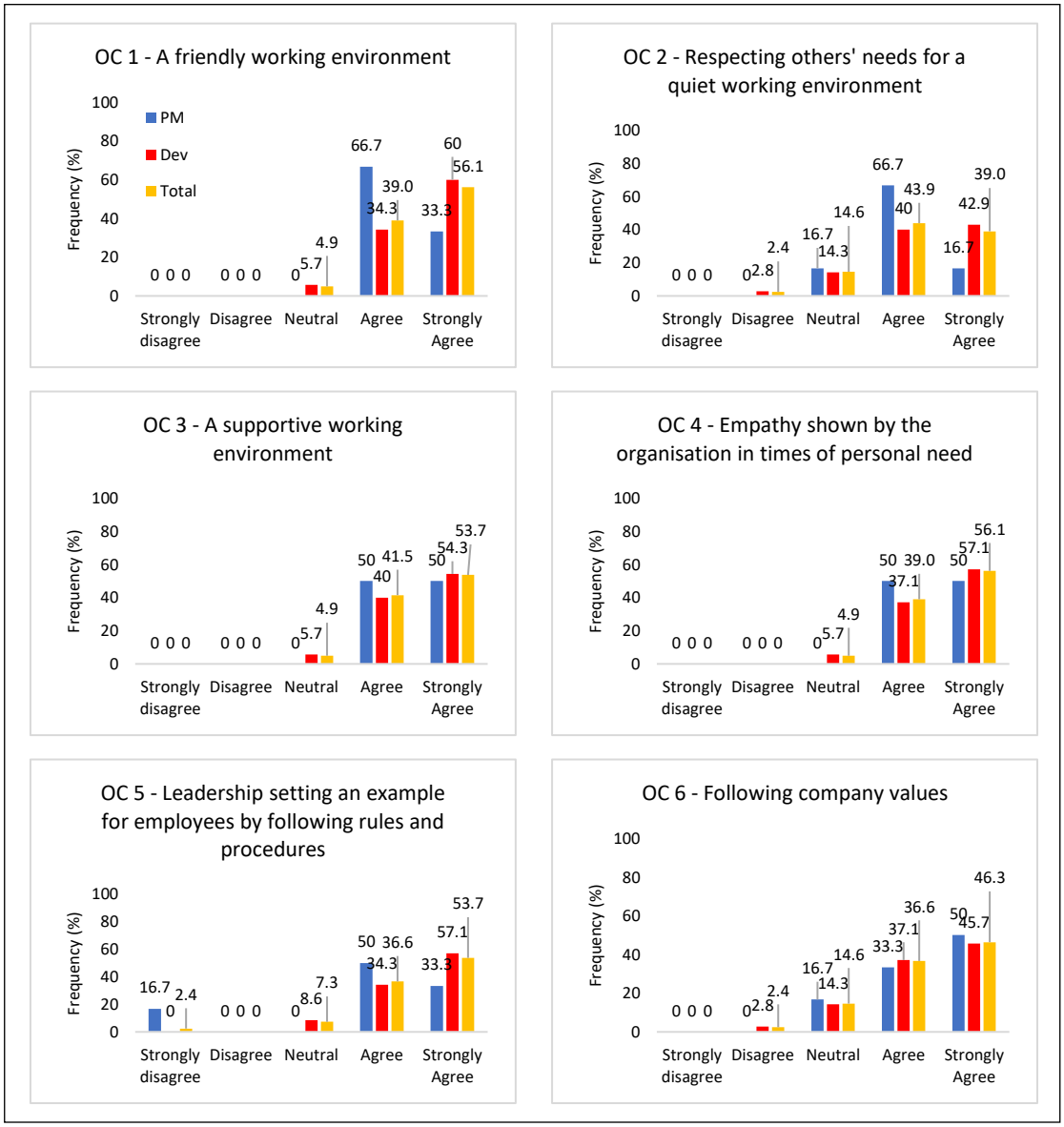

Figure 23: Frequency of responses to whether organisational culture is critical to project success.

4.6.2 Composite measures for organisational factors

In this section, the composite measures for the three organisational factors are presented. Factor analysis was performed to explore the groupings of the items and confirm construct validity (see Table 10). For the SSS, CBO and OC factors, one factor was extracted per construct. To measure the reliability of each composite construct measure, Cronbach's alpha was calculated. For a reliable measure, alpha should exceed 0.7. When calculating reliability on all organisational factors, no items were excluded.

Table 10: Composite measures for organisational factors.

Factor	*Items included*	*Items excluded*	*KMO*	*Resultant Cronbach's alpha*
Support from management and stakeholders	SSS1 – SSS3	-	.644	0.818
Clearly defined business objectives	CBO1 – CBO3	-	.664	0.899
Organisational culture	OC1 – OC6	-	.674	0.817

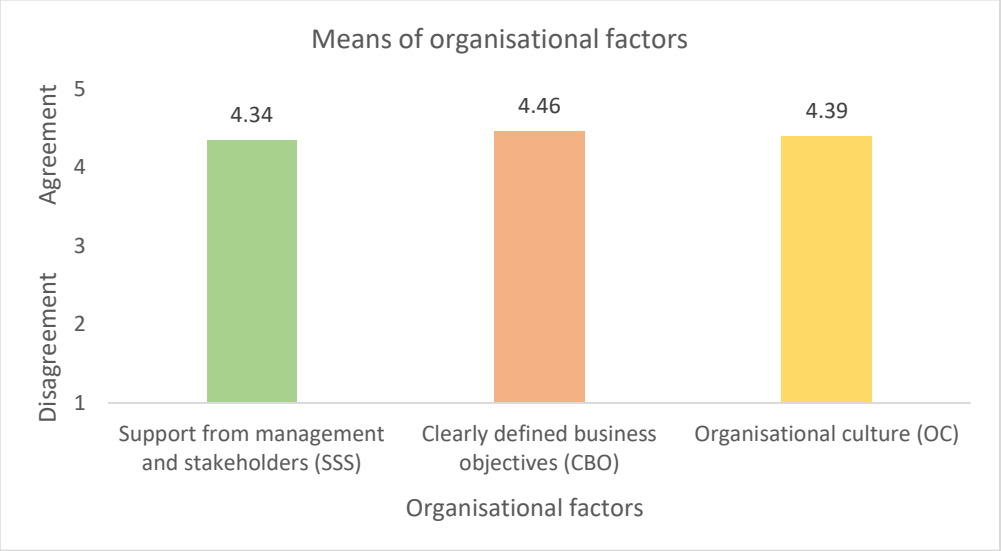

Figure 24: Overall composite means of organisational factors

Figure 24 shows the overall resultant means of all organisational factors. The Wilcoxon signed ranks test was conducted to determine if the average rank of the responses is significantly

different from the central point of the Likert scale (i.e. 3). Support from management and stakeholders emerged as a critical success factor ($Z = -5.469$, $p < .0005$). Clearly defined business objectives are a critical success factor ($Z = -5.662$, $p < .0005$). Organisational culture is also a critical success factor ($Z = -5.607$, $p < .0005$).

A comparison was done to see if results from the organisational factors differed, based on gender, race and role. No significant differences were found. Spearman's correlation tests were conducted to understand if a correlation existed between organisation factors and questions 1.5 to 1.14. A moderately significant correlation was found between the age of respondents and the support from management and stakeholder factor, rho = .327, p=.037. This shows that respondents who are older require more support from management and stakeholders in order for projects to be successful. This may be because older team members are generally in more senior positions and need management support in their supervisory roles. Another correlation was found between support from management and stakeholders and the number of weekends employees worked in a month, rho = .422, p=.006. It appears that employees who tend to work more weekends in a month require more support from management and stakeholders for projects to be successful.

A moderately significant correlation was also identified between organisational culture and the number of weekends employees worked in a month, rho = .474, p=.002. Perhaps employees who work more weekends in a month believe that it is organisational culture to work overtime and on weekends for projects to be successful. This agrees with the results in Table 5. Another correlation was found between organisational culture and the amount of experience software developers should have for projects to be successful, rho = .327, p=.037. People who believe that developers need more years of experience also follow the organisational culture more.

A Friedman's rank test was conducted on all the organisational factors. Support from management and stakeholders had a ranking of 1.87. Clearly defined business objectives had a ranking of 2.16. Organisational culture had a ranking of 1.98. This shows that support from management and stakeholders is considered less critical than other organisational factors. However, there were no significant differences between these three factors, so no factor was significantly less important than any other.

4.6.3 Conclusion on organisational factors

This section analysed the results from organisational factors. Support from stakeholders and management is a significant critical success factor. It was found that project managers consider this factor more critical to project success than do software developers, as their overall means were higher. Project managers and developers believed that more access to support from management and stakeholders, and better communication, will ensure that a project will be successful.

Clearly defined business objectives are critical for success. Project managers also consider this factor more important than do software developers. Organisational culture is also a significant factor for success. Most of the responses ranged between 'neutral' and 'strongly agree', except for responses to OC5 (leadership setting an example for employees by following rules and procedures). One project manager strongly disagreed with this statement. Software developers consider this factor more important than do project managers.

It was found that no one factor was significantly less important than any other. There were also no significant differences between the different genders, races and roles.

4.7 Objective 4: Project factors affecting software project success

The aim of this section was to investigate what project factors are critical to software project success from project managers' and software developers' perspectives. This section investigates nine factors. The results from the questionnaire and interview questions are presented first followed by the results from the composite factors. Figure 25 shows the conceptual framework with the project factors highlighted.

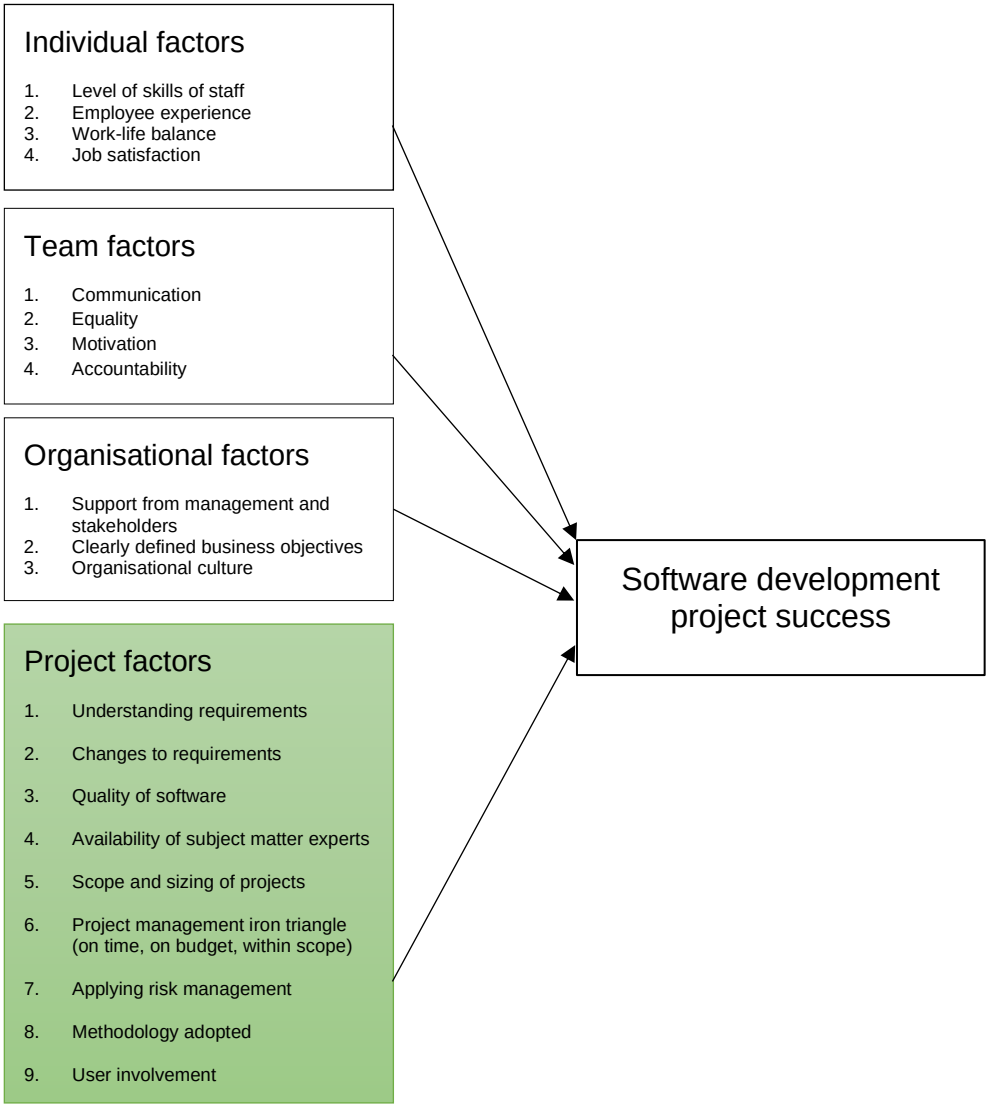

Figure 25: Project factors in the conceptual framework.

4.7.1. Responses to questions on project factors

This section contains the answers from the questionnaire for each of the nine project factors.

4.7.1.1 Understanding requirements

Understanding requirements was the first team factor that was examined. When asked whether understanding requirements is critical to project success, respondents mainly responded with 'neutral', 'agree' and 'strongly agree' (see Figure 26).

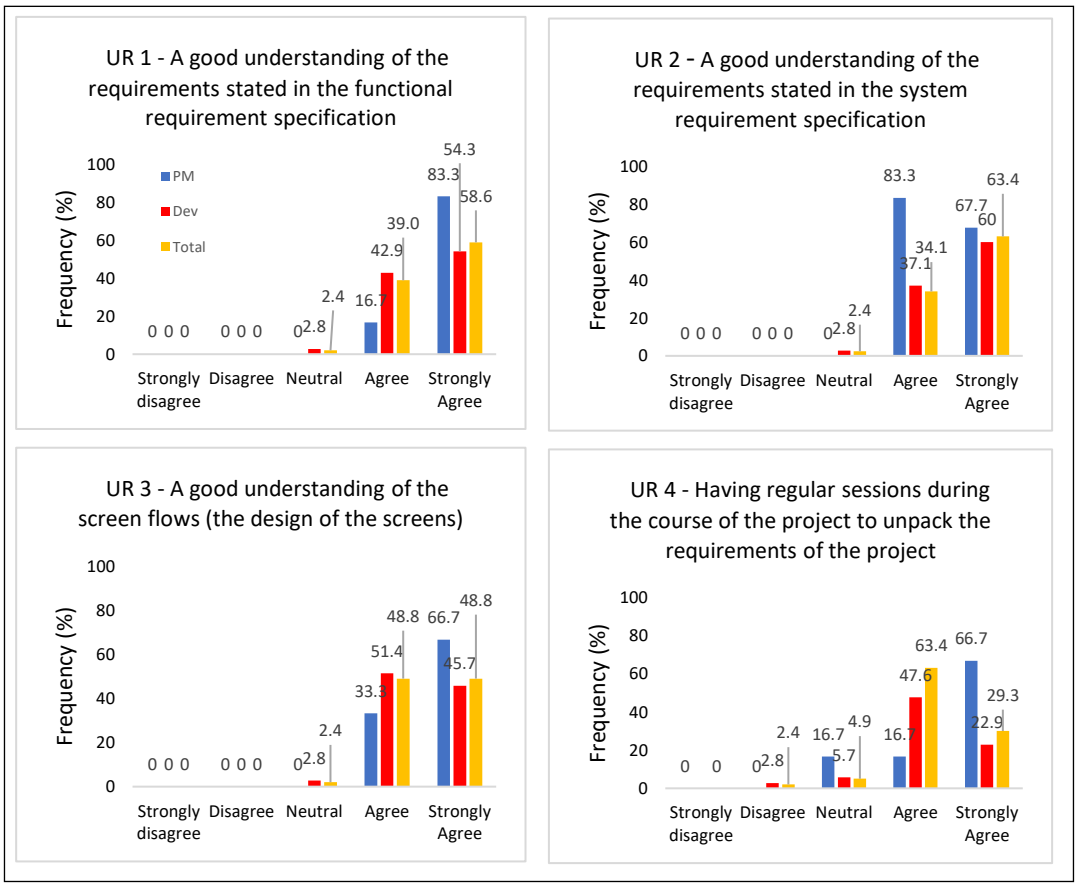

Figure 26: Frequency of responses to whether understanding requirements is critical to project success.

The Wilcoxon signed rank test is used to determine if the average rank of the responses is significantly different from the central point of the Likert scale (i.e. 3). There was no significant disagreement with any of the variables for the understanding requirements questions (see Figure 26). A good understanding of the requirements stated in the functional requirement specification is critical to project success (Z= -5.706, p<.0005). A good understanding of the requirements

stated in the system requirements specification is critical to project success (Z=-5.734, p<.0005). A good understanding of the screen flows is critical to project success (Z=-5.684, p<.0005). Having regular sessions during the project to unpack the requirements of the project is critical to project success (Z=-5.491, p<.0005). It is interesting to note that not all respondents believed that having regular sessions to unpack requirements is important, as a total of 2.8% of respondents (all developers) disagreed with the statement and 4.9% (a mixture of project managers and developers) were neutral towards the statement.

4.7.1.2 Changes to requirements

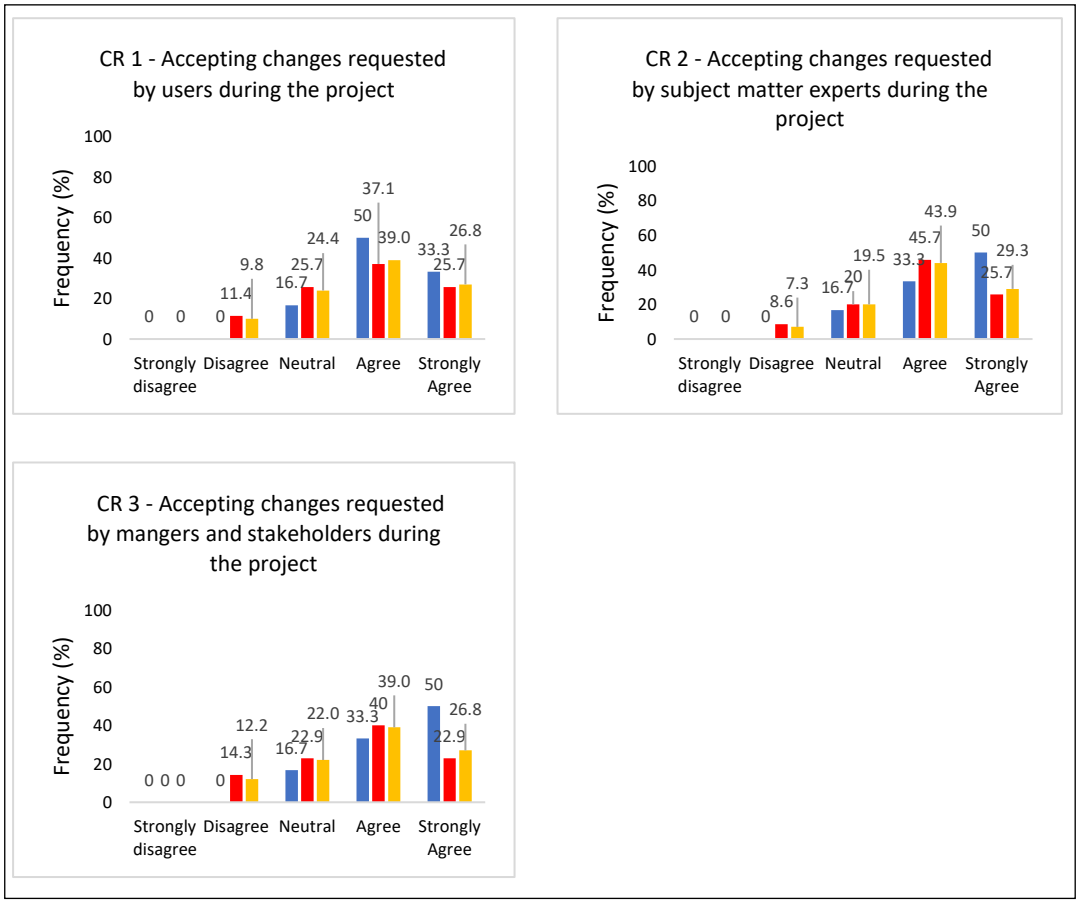

Figure 27: Frequency of responses to whether changing requirements is critical to project success.

Figure 27 shows responses to whether accepting changes are critical to project success. The Wilcoxon signed ranks test was conducted to determine if the average rank of the responses is significantly different from the central point of the Likert scale (i.e. 3). There was no significant disagreement that changes to requirements are critical to project success. Accepting changes requested by users is critical to project success (Z= -4.196, p<.0005). Accepting changes

requested by subject matter experts during the project is critical to project success (Z= -4.593, p<.0005). Accepting changes requested by managers and stakeholders is critical to project success (Z= -4.068, p<.0005). However, some software developers disagreed with accepting changes from anyone during the software development lifecycle (see Figure 27).

Table 11: Means and standard deviations of the responses of project managers and software developers to changing requirements

	Project managers	Software developers
	Mean (SD)	Mean (SD)
CR1 - Accepting changes requested by users during the project	4.2 (0.8)	3.8 (1)
CR2 - Accepting changes requested by the subject matter experts	4.3 (0.8)	3.9 (0.9)
CR3 - Accepting changes requested by stakeholders and management	4.3 (0.8)	3.7 (1)

When the results for accepting changes in requirements were analysed, it was seen that project managers and software developers had similar attitudes to accepting changes (see Figure 27). Project managers agreed with accepting changes from users, subject matter experts and stakeholders and management, as indicated by the means ranging between 4.2 and 4.3 (see Table 11). Software developers also seemed to agree. However, the lower means ranged between 3.7 and 3.9. Even though most software developers agreed that accepting changes to requirements was critical to success, some of them responded with neutrality or disagreement (see Figure 27). Developers are less enthusiastic about changing requirements during the project than project managers. Developers seem more likely to accept changes, in decreasing order, from subject matter experts, users and stakeholders and managers. By comparison, project managers seem equally likely to accept changes to requirements from subject matter experts, stakeholders and mangers, followed by users.

4.7.1.3 Quality control

The quality control factor was investigated through six statements (see Figure 28).

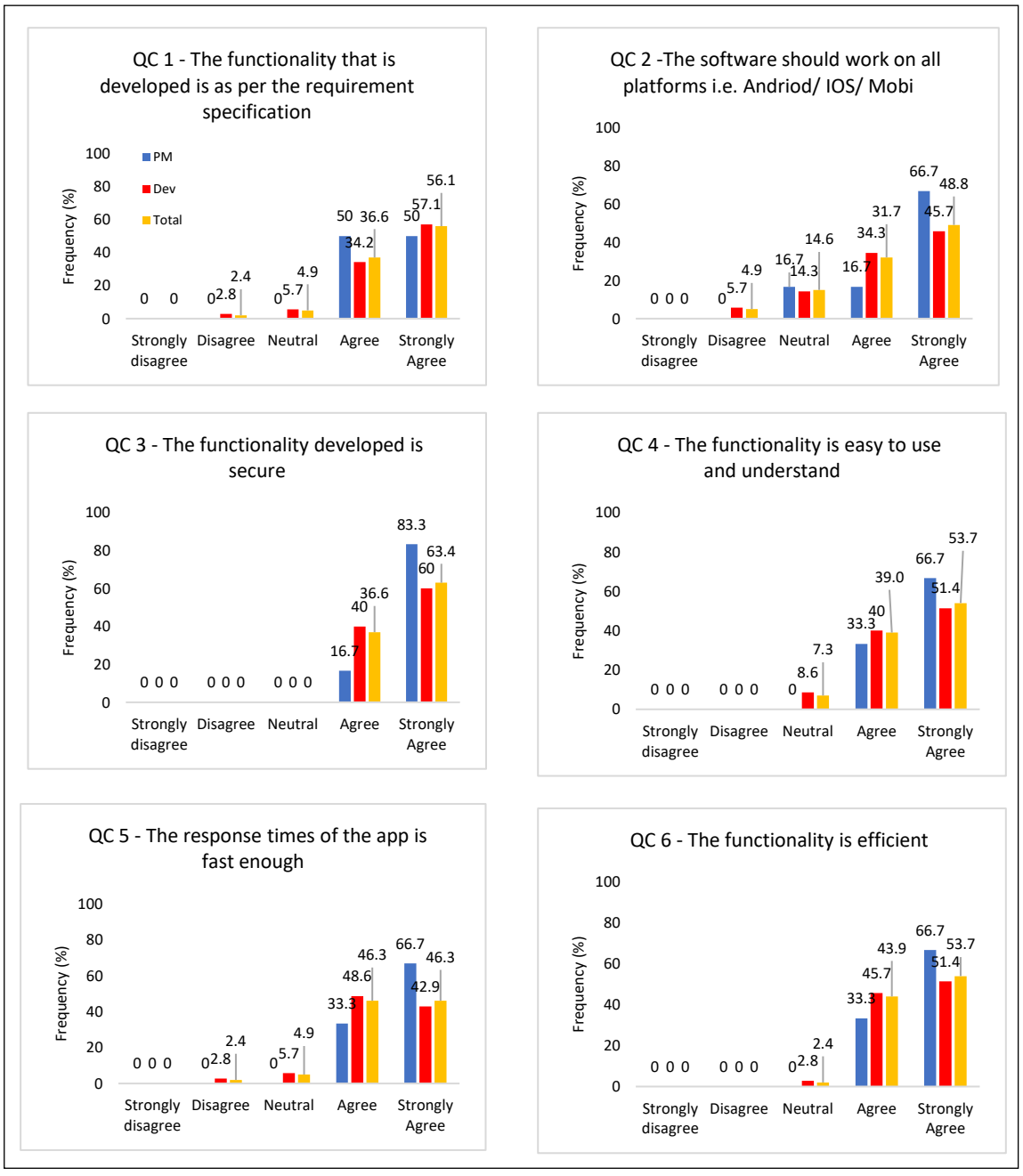

Figure 28: Frequency of responses to whether quality control is critical to project success

The results for quality control show that no significant disagreement exists based on the Wilcoxon signed rank test results. The functionality is developed as per the requirement specification and is critical to project success (Z= -5.508, p<.0005). Even though this is a critical success factor,

one software developer (2.8%) disagreed with this statement. This software developer had between four and eight years of experience. The fact that the software should work on all platforms is critical to project success (Z= -5.060, p<.0005). This question also showed that two software developers disagreed (5.7%). Respondents either agreed or strongly agreed with the statement (Z= -5.759, p<.0005) that the functionality that is developed is secure and critical to project success. That the functionality is easy to use and understand is critical to project success (Z= -5.555, p<.0005). That the response times of the app are fast enough also proved to be a critical success factor (Z= -5.463, p<.0005). However, one software developer (2.8%) disagreed with the statement that efficient functionality is a critical success factor (Z= -5.690, p<.0005).

4.7.1.4 Availability of subject matter experts

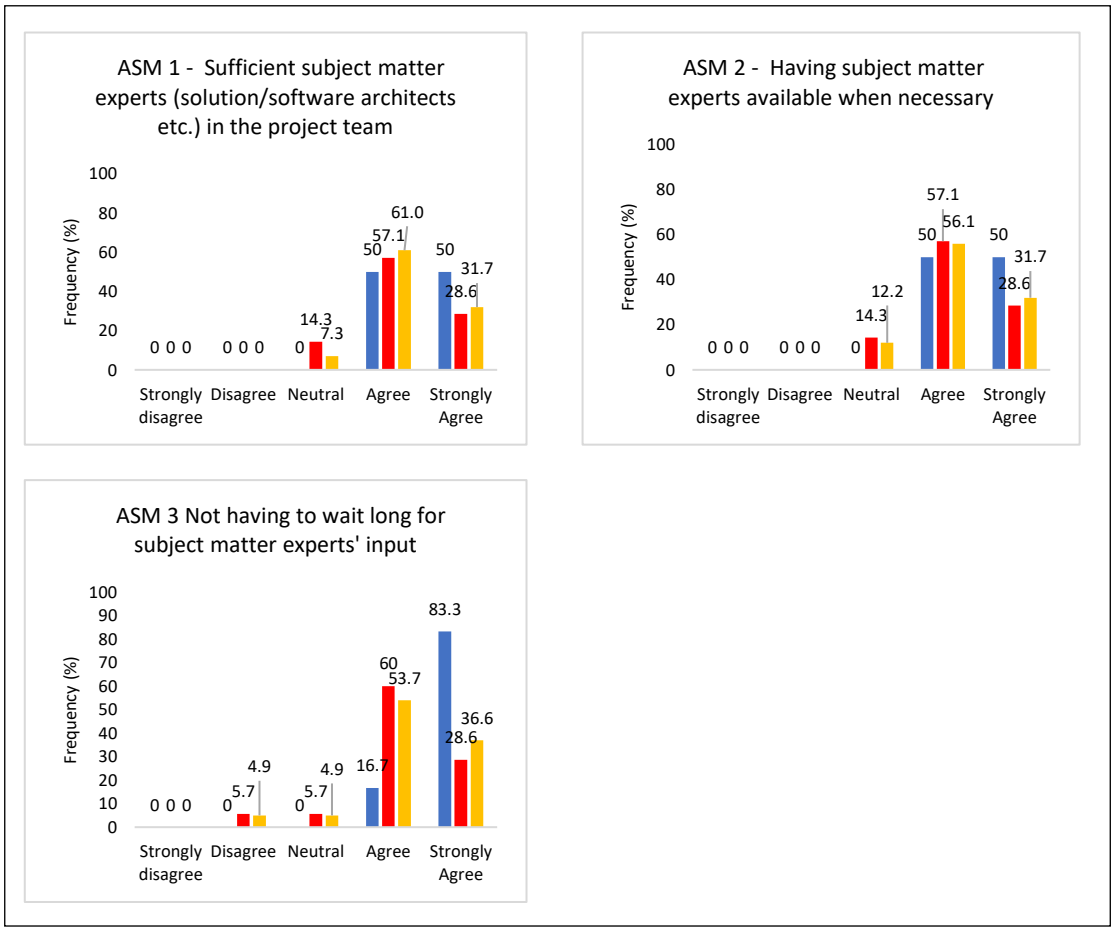

Figure 29: Frequency of responses to whether availability of subject matter experts is critical to project success.

In this study, subject matter experts include software and solutions architects. The results for this factor show that the availability of subject matter experts is critical to project success. The Wilcoxon signed ranks test was conducted to determine if the average rank of the responses is significantly different from the central point of the Likert scale (i.e. 3). Having enough subject matter experts in the project team is critical to project success ($Z = -5.596$, $p < .0005$). Having subject matter experts available when necessary is critical to project success ($Z = -5.436$, $p < .0005$). Not having to wait long for subject matter experts' input is critical to project success ($Z = -5.281$, $p < .0005$). There is no significant disagreement with the questions about the availability of subject matter experts. Figure 28 shows that project managers seem to agree more that the availability of subject matter experts is critical to success. Project managers mainly responded with 'agree' and 'strongly agree'. Whilst it can be seen that majority of software developers agree with the statements under this factor, not all of them believe having sufficient subject matter experts in the team results in project success, as 14.3% responded with neutrality; and 14.3% of software developers also responded with neutrality when asked whether having subject matter experts available when necessary is critical to project success. Of the software developers, 5.7% also disagreed that not having to wait long for subject matter experts' input is critical to project success.

4.7.1.5 Scope and sizing of projects

Figure 30 shows the responses from project managers and software developers on whether the scope and size of a project is critical to project success.

There was no significant disagreement with any of the questions for scope and sizing of projects based on the Wilcoxon signed rank tests results. Sizing projects accurately is a critical success factor ($Z = -5.642$, $p < .0005$). Projects that are accurately scoped is critical to success ($Z = -5.752$, $p < .0005$). Projects with smaller scope and size are critical to project success ($Z = -3.604$, $p < .0005$). Even though many of the responses agreed with this statement, it is evident in Figure 28 that there were respondents who disagreed that projects with a smaller scope and size would be successful. It is also interesting to note that this variable seemed to have the highest number of neutral responses. Breaking projects into smaller phases can also be considered a critical success factor ($Z = -4.237$, $p < .0005$). However, there were some respondents who disagreed with this statement. It is interesting to note that the one developer who strongly disagreed with SSP3 and SSP4 had less than four years of experience. Lastly, it is critical to project success if projects can be completed in a realistic time ($Z = -5.617$, $p < .0005$).

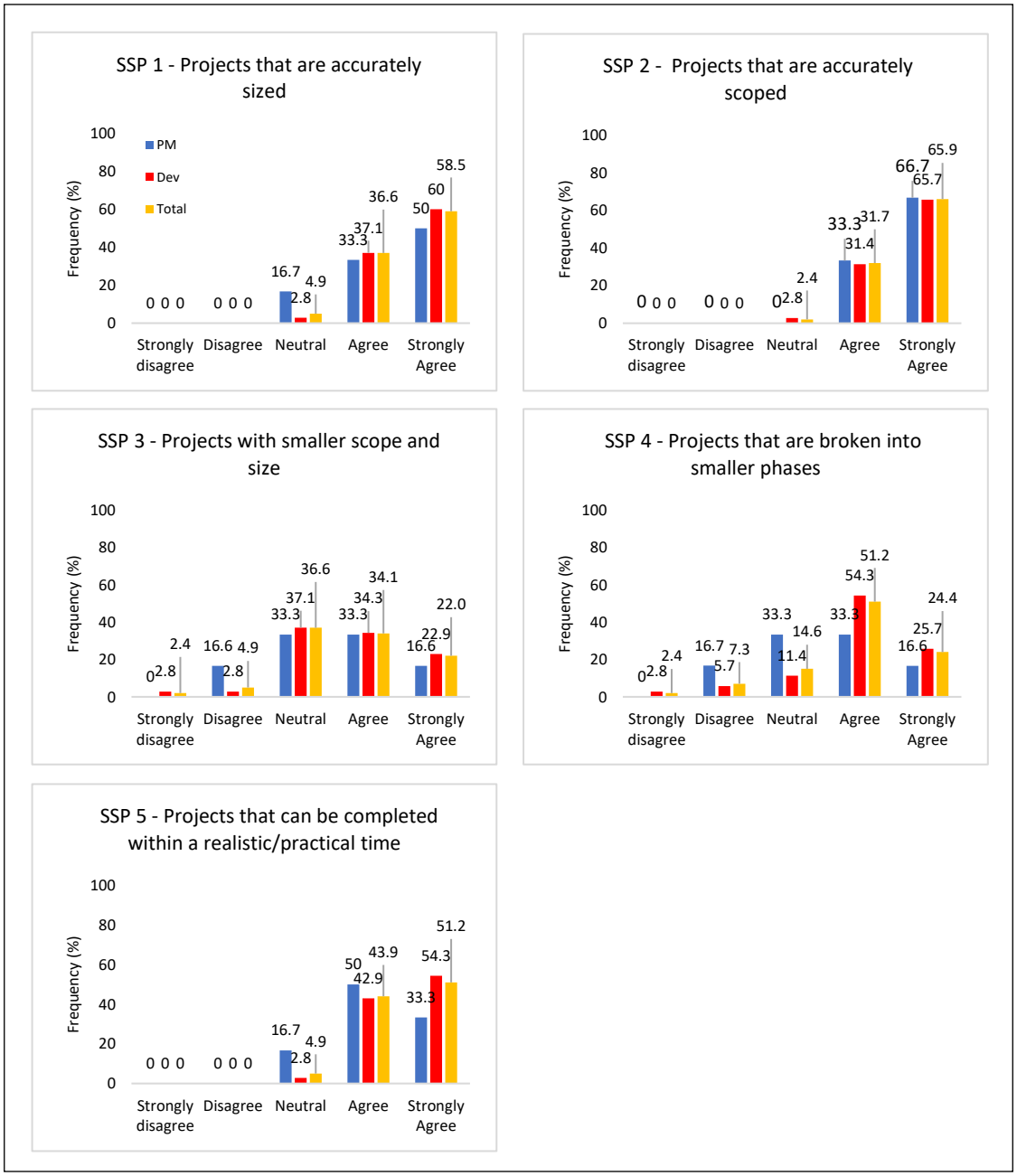

Figure 30: Frequency of responses to whether scope and sizing of projects is critical to project success.

4.7.1.6 The project management iron triangle

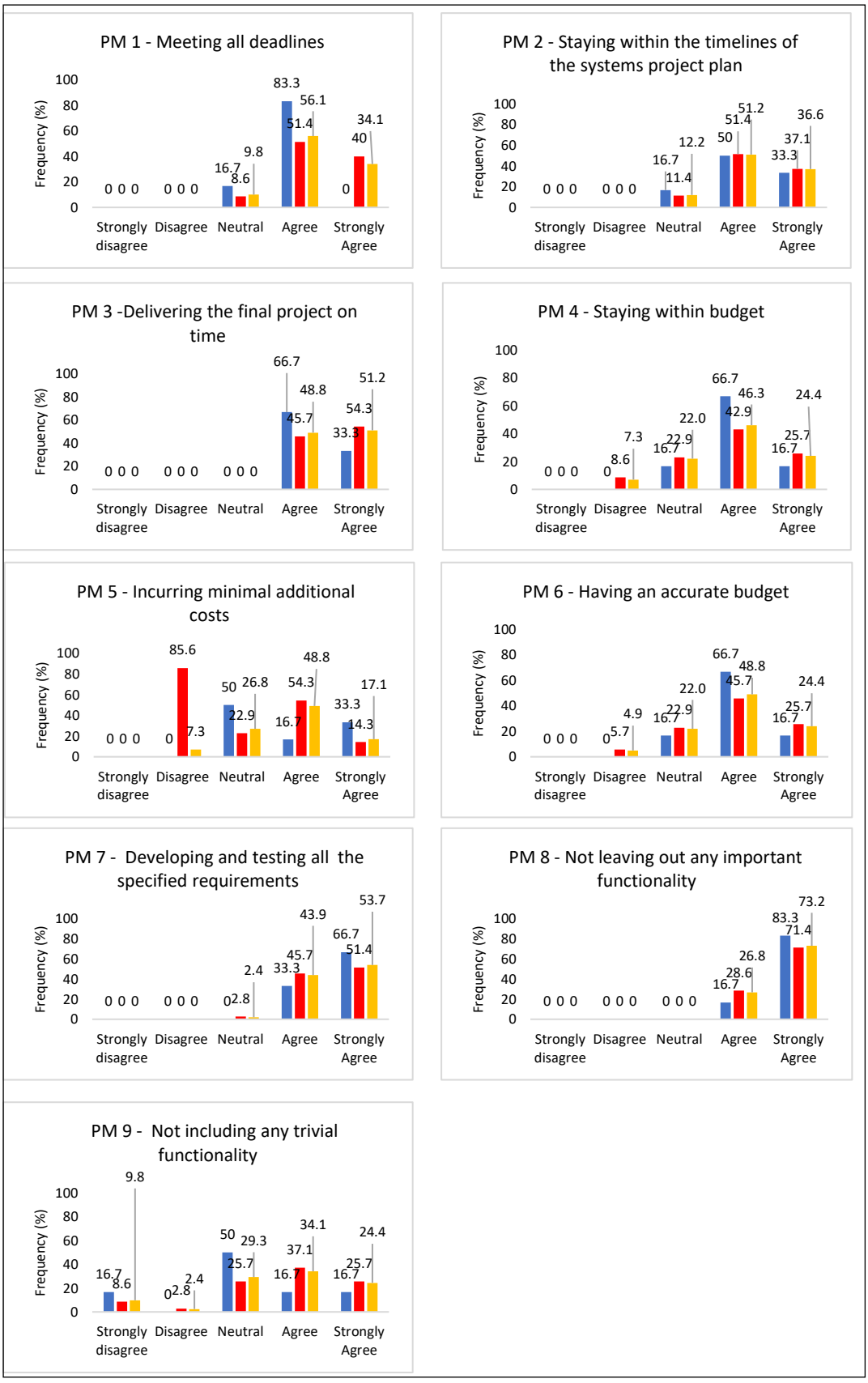

Figure 31: Frequency of responses to whether project management factors are critical to project success.

Figure 31 shows how respondents felt about projects being on time, within budget and within the defined scope. The Wilcoxon signed ranks test was conducted to determine if the average rank of the responses is significantly different from the central point of the Likert scale (i.e. 3). Meeting all deadlines is a critical success factor (Z= -5.500, p<.0005). Staying within the timelines of the systems project plan is a significant success factor (Z= -5.410, p<.0005). Delivering the final project on time is a critical success factor, as both project managers and software developers responded with either 'agree' or 'strongly agree' (Z= -5.755, p<.0005).

Staying within budget is a critical success factor (Z= -4.485, p<.0005). Project managers seem to agree more than do developers that staying within the budget is critical to project success. Whilst incurring minimal additional costs is also seen as a critical success factor (Z= -4.284, p<.0005), 85.6% of software developers disagreed with this. This may be because software developers do not have sight of the costs that go into a project. Having an accurate budget is a critical success factor (Z= -4.710, p<.0005). Developing and testing all of the specified requirements is a critical success factor (Z= -5.690, p < .0005). Not leaving out any important functionality produced results of mainly 'agree' and 'strongly agree', indicating that it is a critical success factor (Z= -5.877, p < .0005). There was significant agreement that not including any trivial functionality is critical to project success (Z= -2.664, p = .008).

PM7 and PM8 (see Figure 31) were compared with the results from the understanding requirements factor in Section 4.7.1.2. Project managers agreed and strongly agreed with UR1 (a good understanding of the requirements stated in the functional requirement specification), UR2 (a good understanding of the requirements stated in the system requirement specification) and UR3 (a good understanding of the screen flows). In the same way, project managers agreed and strongly agreed with PM7 and PM8. Most software developers also agreed and strongly agreed with UR1, UR2 and UR3. However, one developer responded with neutrality to all three questions. Most software developers agreed and strongly agreed with PM7 and PM8; however, one developer was neutral. It is interesting to note that the developer who responded with neutrality to PM7 also responded with neutrality to UR3.

4.7.1.7 Risk management

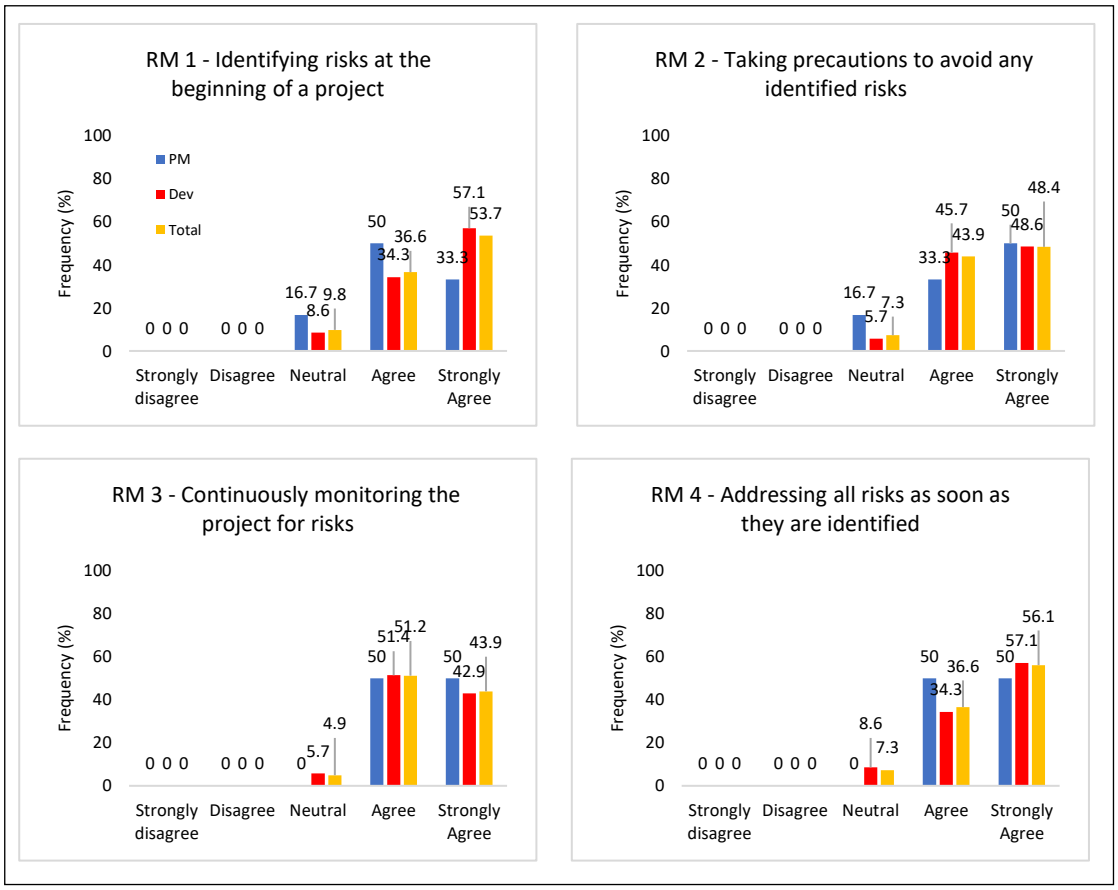

Figure 32: Frequency of responses to whether risk management factors are critical to project success.

It is important that risks are identified and mitigated. This can be seen in Figure 32, as most respondents responded with 'neutral', 'agree' and 'strongly agree'. Identifying risks at the beginning of the project is a critical success factor as 53.7% of respondents strongly agreed with the statement ($Z = -5.488$, $p < .0005$). Of the respondents, 48.8% strongly agreed that taking precautions to avoid the risks identified is also critical to project success ($Z = -5.543$, $p < .0005$). Continuously monitoring the project for risks contributes to success, as 51.2% of respondents agreed with the statement ($Z = -5.617$, $p < .0005$). Lastly, 56.1% of respondents believe that addressing risks as soon as they are identified is critical to project success ($Z = -5.565$, $p < .0005$). It is worth noting that no respondents strongly disagreed or disagreed with any of these statements.

4.7.1.8 Methodology adopted

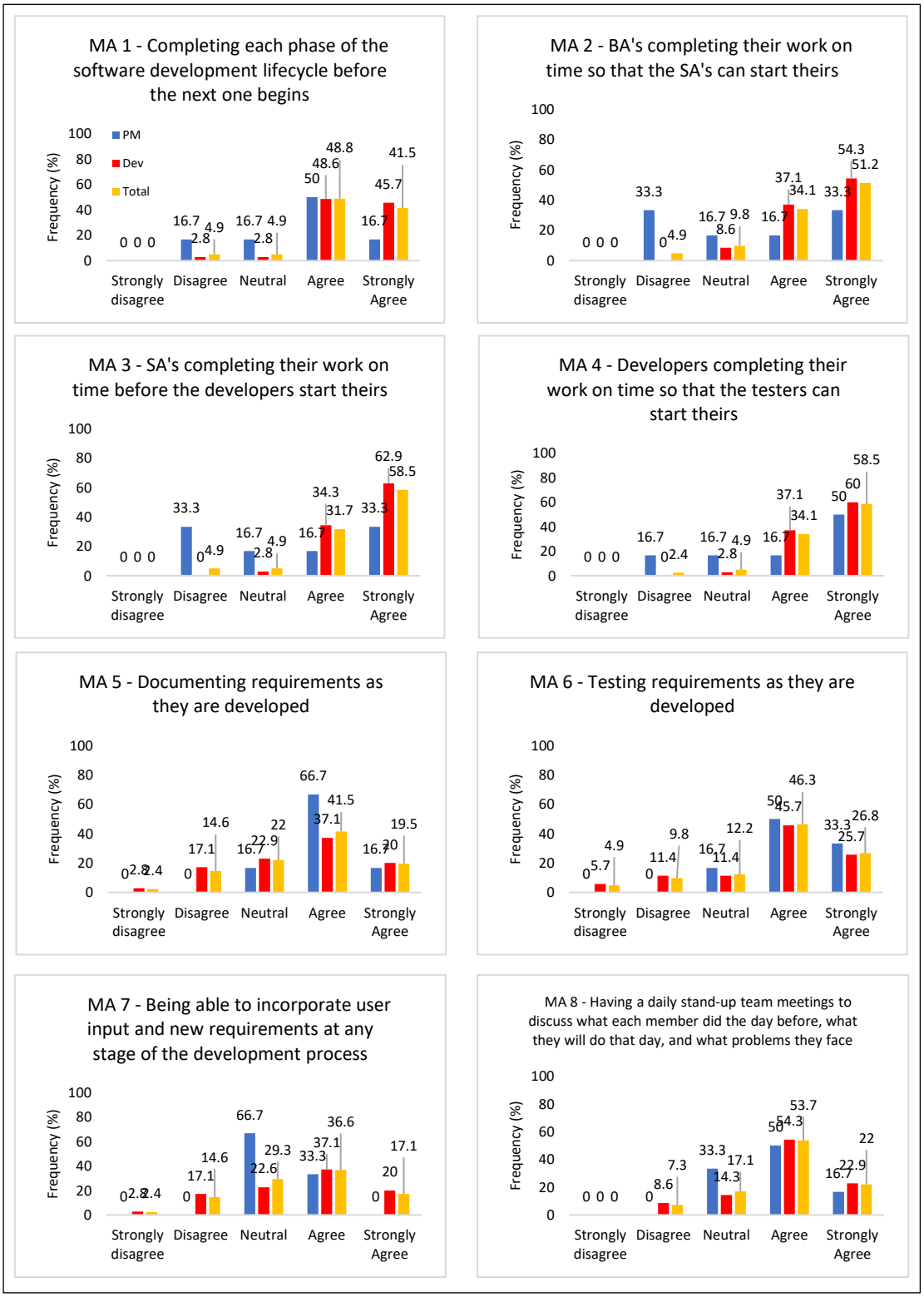

Figure 33: Frequency of responses to whether the methodology adopted is critical to project success.

Figure 33 shows responses to what type of methodology contributes to project success. Respondents were asked to indicate their level of agreement to statements about traditional (waterfall) and agile approaches (MA1 - 4 and MA5 – 8, respectively). The results varied and it seems as if respondents like both approaches. Based on the Wilcoxon signed rank test results, completing each phase of the software development lifecycle before the next can begin is a critical success factor ($Z= -5.291$, $p<.0005$). One project manager and one software developer seem to disagree with this statement. BAs completing their work on time so that SAs can begin theirs is critical to project success ($Z= -5.214$, $p<.0005$). SAs completing their work on time before developers start theirs is a critical success factor ($Z= -5.411$, $p<.0005$). Developers completing their work on time so that testers can start theirs is critical to project success ($Z= -5.527$, $p<.0005$). Based on Figure 33, it can be seen that project managers disagree that having professionals completing their work before the next can begin theirs is critical to project success. Software developers seem to agree more that everyone in the team needs to complete their work before the next team members can begin theirs.

Documenting requirements as they are developed is critical to project success ($Z= -2.864$, $p=.004$). Project managers agree, more than developers, that documenting requirements as they are developed is critical to project success. Testing requirements as they are developed is critical to project success ($Z= -4.657$, $p<.0005$). Some developers do not agree that this is a critical success factor. Being able to incorporate user input and new requirements at any stage of the development process is critical to project success ($Z= -3.221$, $p=.001$). It is worth noting that this question resulted in the highest neutral responses by project managers. Software developers also showed their disagreement with this statement. Having daily stand-up meetings is critical to project success ($Z= -3.673$, $p<.0005$), although 8.6% of developers seem to disagree to this question.

Table 12: Means and standard deviations of the responses on the methodology adopted.

	Project managers	Software developers
	Mean (SD)	Mean (SD)
MA1- Completing each phase of the software development lifecycle before the next one begins	3.7 (0.8)	4.4 (0.7)
MA2 - BA's completing their work on time so that the SA's can start theirs	3.5 (0.8)	4.5 (0.7)

MA3 - SA's completing their work on time before the developers start theirs	4.4 (0.7)	4.6 (0.6)
MA4 - Developers completing their work on time so that the testers can start theirs	4.0 (0.7)	4.6 (0.6)
MA5 - Documenting requirements as they are developed	3.5 (1.1)	3.5 (1.0)
MA6 - Testing requirements as they are developed	3.9 (0.8)	3.9 (1.1)
MA7 - Being able to incorporate user input and new requirements at any stage of the development process	3.6 (1.0)	3.5 (1.1)
MA8 - Having a daily stand-up team meetings to discuss what each member did the day before, what they will do that day, and what problems they face	3.8 (1.1)	3.7 (1.1)

Table 12 shows the means and standard deviations of responses to the methodology factor. The responses differ. Software developers' means are much higher for questions related to structured approaches, indicating that developers believe more in a structured approach. Questions relating to flexible approaches (MA5 – MA8) had varying results. Project managers' and software developers' perceptions on documenting requirements as they are developed and testing requirements as they are developed were very similar (see Table 12). Software developers do not particularly believe that incorporating user input or requirements during the project is critical to success; whereas project managers do. Software developers also do not agree that having daily stand-up meetings is critical to project success.

The standard deviations of MA5 to MA8 are much higher than those of MA1 to MA4. This indicates a larger spread of responses from project managers and software developers. Stand-up meetings are a way of keeping track of projects using the 'scrum' agile approach. Question MA8 reflected disagreement from developers, which could be as a result of the team not working in an agile manner or using the scrum approach. MA7 asked respondents about accepting changes during the project, so the results were compared to those on the changes to requirements factor in Section 4.7.1.2. Project managers mainly responded with 'neutral' to MA7. However, in CR1 (accepting changes requested by users during the project), CR2 (accepting changes requested by subject matter experts during the project) and CR3 (accepting changes requested by managers and stakeholders during the project), only one project manager responded with 'neutral', while the rest of the project managers either agreed or strongly agreed with the statements. Software developers' responses seemed to range from 'strongly disagree' to 'strongly agree' for MA7. However, for CR1, CR2 and CR3 the responses ranged between 'disagree' and 'strongly agree'.

For CR1, CR2 and CR3, developers are less keen on accepting changes to requirements than project managers are (see Table 12). However, for MA7, project managers' and developers' responses (mean values) were more similar (see Table 12), and both groups seem to be less keen on accepting new requirements.

4.7.1.9 User involvement

Respondents were asked to indicate their agreement on whether having users involved in the project contributes to software project success. Figure 34 shows the results.

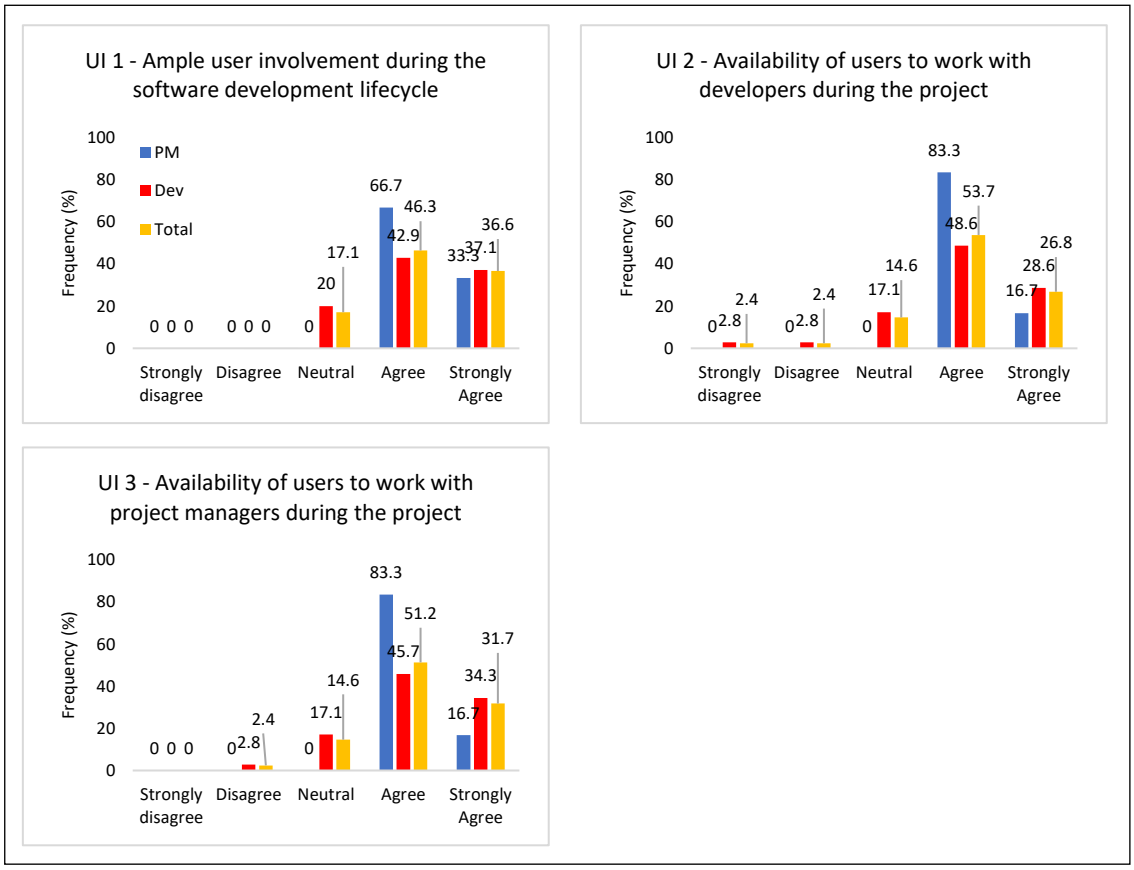

Figure 34: Frequency of responses to whether user involvement is critical to project success.

Ample user involvement during the software development lifecycle is critical to project success ($Z = -5.252$, $p < .0005$). Project managers were unanimous in their agreement. However, some software developers responded with 'neutral'. Availability of users to work with developers during the project is critical to project success ($Z = -4.664$, $p < .0005$). It is interesting to note that all project managers 'agreed' and 'strongly agreed' that users working with developers during a project will promote project success. However, some developers disagreed, and some responded with 'neutral'. Availability of users to work with project managers during the project is also a

critical success factor (Z= -5.159, p<.0005). All project managers were in agreement that this is necessary for projects to be successful. It is worth pointing out that project managers responded in the exact same way to UI2 and UI3.

4.7.2 Composite measures for project factors

In this section, the composite measures for the all the project factors are presented. Factor analysis was performed on all factors and the project management iron triangle and the approach to development (traditional or agile) produced further factors.

4.7.2.1 Factor analysis for sub factors

Project management iron triangle

Factor analysis was conducted on the project management iron triangle sub factors, projects being on brief, on time and on budget. The three factors were extracted which account for 76.24% of the variance in the data, as seen in Table 14 below.

Table 13: Factor analysis on project management iron triangle questions.

	Factor		
	On time	On budget	On brief
TF PM1 Meeting all deadlines	0.908		
TF PM2 Staying within the timelines of the system's project plan	0.868		
TF PM3 Delivering the final project on time	0.602		
TF PM5 Incurring minimal additional costs		0.991	
TF PM6 Having an accurate budget		0.823	
TF PM4 Staying within budget		0.624	
TF PM8 Not leaving out any important functionality			0.812
TF PM7 Developing and testing all the specified requirements			0.379

These sub factors were tested for reliability using Cronbach's alpha. An alpha >.7 was considered to indicate a reliable measure. PM9 (not leaving out any trivial functionality) was excluded from the 'on brief' factor, as it could be considered very similar to PM8. The alpha value for the 'on brief' factor is rather low, .527.

Table 14: The sub factors produced from the project management iron triangle.

Sub Factor No.	*Sub factor*	*KMO*	*Reliability (Cronbach's alpha)*
Sub Factor 1	On time	.759	.822
Sub Factor 2	On budget	.759	.845
Sub Factor 3	On brief	.759	.527

Methodology adopted

Factor analysis was conducted on the methodology adopted factor (see Table16). It can be seen that two sub-factors were identified: the structured (traditional) and flexible (agile) approaches to systems development. These two factors account for 65.69% of variance.

Table 15: Factor analysis on methodology adopted.

	Factor	
	Traditional	Agile
MA3 - SA's completing their work on time before the developers start theirs	0.963	
MA2 - BA's completing their work on time so that the SA's can start theirs	0.879	
MA4 - Developers completing their work on time so that the testers can start theirs	0.77	
MA1 - Completing each phase of the software development lifecycle before the next one begins	0.52	

MA7 - Being able to incorporate user input and new requirements at any stage of the development process	0.787
MA5 - Documenting requirements as they are developed	0.682
MA8 - Having daily stand up meetings	0.603
MA6 - Testing requirements as they are developed	0.592

Table 16: The factors produced from the methodology adopted.

Sub Factor No	*Sub factor*	*KMO*	*Reliability (Cronbach's alpha)*
Sub Factor 1	Traditional	.682	.854
Sub Factor 2	Agile	.682	.749

4.7.2.2 Composite measures for project factors

In this section, the composite measures for the project factors are presented. Factor analysis was performed to explore the groupings of the items and confirm construct validity. For the CR, QC, RM, and UI one factor was extracted per construct. However, for the UR, ASM and SP factors constructs, results from a factor analysis were not acceptable as the KMO value was less than 0.6, indicating that the data was not adequate for successful factor extraction to take place. To measure the reliability of each composite construct measure, Cronbach's alpha was calculated. For a reliable measure, alpha should exceed 0.7. One question had to be excluded from each of the 'availability of subject matter experts' factor, the scope and sizing of projects and the project management iron triangle because the item-total correlation was too low. ASM1 (sufficient subject matter experts in a project team) was excluded from the availability of subject matter experts' composite value. SSP3 (projects with smaller scope and size), SSP4 (projects that are broken into smaller phases) and SSP5 (projects that can be completed within a realistic or practical time) were excluded from the scope and sizing of projects' composite value. PM9 (not including any trivial functionality) was excluded from the project management iron triangle composite value.

Table 17: Composite measures for project factors.

Factor	*Items included*	*Items excluded*	*KMO*	*Resultant Cronbach's alpha*	*Z*	*p*
Understanding requirements	UR1 – UR4	-		0.777	-5.613	<.0005
Changes to requirements	CR1 – CR3	-	.634	0.763	-4.726	<.0005
Software quality	QC1 – QC6	-	.668	0.778	-5.606	<.0005
Availability of subject matter experts	ASM2 – ASM3	ASM1		0.766	-5.474	<.0005
Scope and sizing of projects	SP1 – SP2	SP3 – SP5		0.826	-5.667	<.0005
Project management iron triangle – On time	PM1 – PM3	-	.759	0.822	-2.007	<.0005
Project management iron triangle - On budget	PM4 – PM6	-	.759	0.845	-2.228	.026
Project management iron triangle - On brief	PM7 – PM8	PM9	.759	0.527	-3.860	<.0005
Risk management	RM1 – RM4	-	.756	0.859	-5.623	<.0005
Methodology adopted – Traditional	MA1 – MA4	-	.682	0.854	-5.436	<.0005
Methodology adopted - Agile	MA5 – MA8	-	.682	0.749	-4.486	<.0005
User involvement	UI1 – UI3	-	.709	0.829	-5.396	<.0005

Figure 35 shows the overall resultant means for each team factor. All factors are significant (see Table 18 for Z and p values). A comparison was done to investigate if the results differed, based on the gender of participants. No significant difference was found. Analysis was also done to see if results differed based on the race of each participant. Once again, no significant difference was recorded. A final comparison was conducted to see if results differed based on the participants' roles. No significant difference was found.

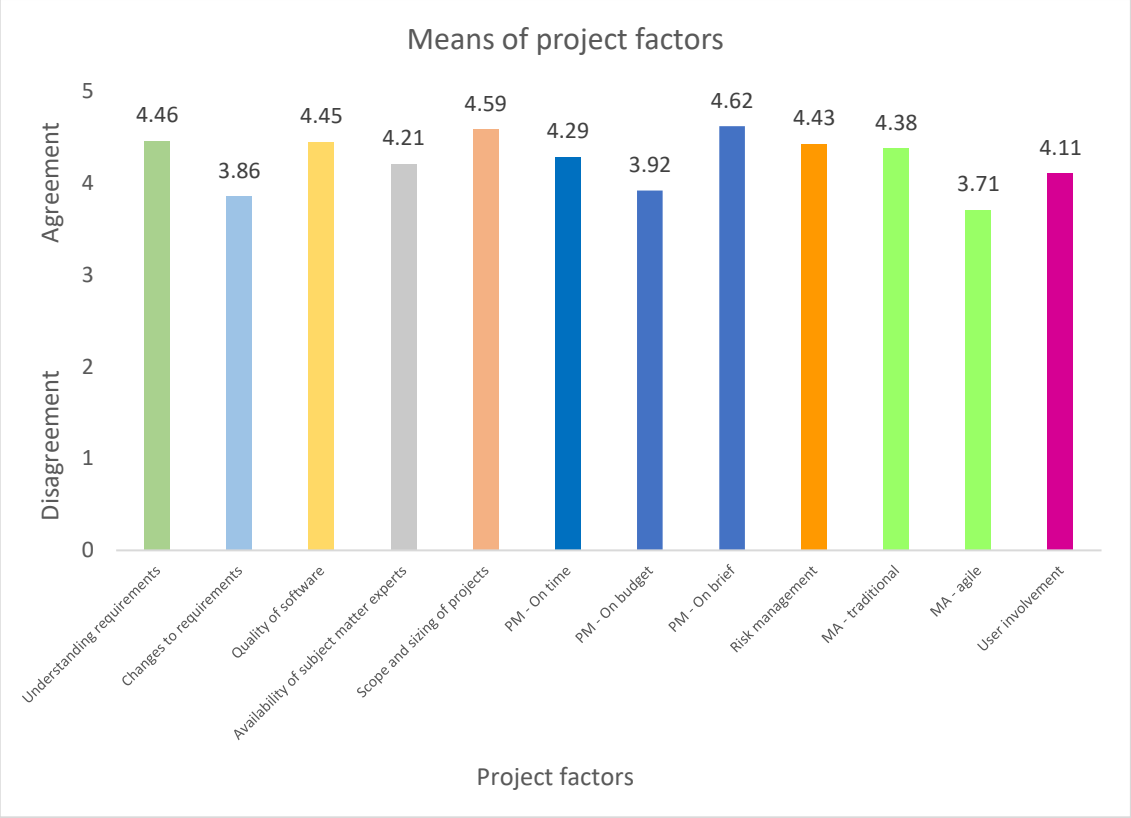

Figure 35: Overall composite means to responses on project factors.

Further analysis was conducted (Spearman's test) to see if there existed a correlation between team factors and questions 1.5 to 1.14. A moderately significant correlation existed between the number of hours each employee works per week and agreement with the statement that changes to requirements during the project is critical to success, rho = .341, p = .029. It can be noted that accepting more changes during a project, increases the number of hours employees work in a week.

A Friedman's test was executed to see how each of the team factors ranked (see Table 19). Within the team factors, there are significant differences in perceptions of how critical the 12 factors are to project success, X^2=119.169, p<.0005. The highest ranked factor was the project management iron triangle, 'on brief'. This indicates that this factor is most critical to project success, closely followed by the scope and sizing of projects and quality control. It can be noted that the methodology adopted, the agile development factor (MA5 – MA8), project management, on budget and changes to requirements were ranked significantly lower than the rest of the factors (significance ranges from p<.0005 to p=.026).

Table 18: Ranking of project factors.

	Factors	Mean Rank	Significantly more critical to software development project success than:
1	Project management – on brief (PM7 – 8)	10.74	QC, UR, MA - Traditional, RM, PM - On time, ASM, UI, CR, PM - On budget, MA - Agile
2	Scope and sizing of projects (SSP)	10.28	PM - On time, ASM, UI, CR, PM - On budget, MA - Agile
3	Quality control (QC)	9.44	ASM, UI, CR, PM - On budget, MA - Agile
4	Understanding requirements (UR)	9.10	ASM, UI, CR, PM - On budget, MA - Agile
5	Methodology adopted – traditional (MA1 - 4)	9.09	UI, CR, PM - On budget, MA - Agile
6	Risk management (RM)	9.00	UI, CR, PM - On budget, MA - Agile
7	Project management – on time (PM1 – 3)	8.04	CR, PM - On budget, MA - Agile
8	Availability of subject matter experts (ASM)	7.12	PM - On budget, MA - Agile
9	User involvement (UI)	7.11	PM - On budget, MA - Agile
10	Changes to requirements (CR)	5.77	
11	Project management – on budget (PM4 – 6)	5.05	
12	Methodology adopted – agile (MA5 – 8)	4.61	

Table 19 also shows which factors are more significant than others. The factor that proved to be most significant was project management ‘on brief’. In it, all p values ranged between 0.000 and 0.037, making it more significant than 13 other factors (see first row of Table 19). The other two factors in the project management iron triangle, ‘on time’ and ‘on budget’ ranked at position nine and position 14. The factor for on time was more significant than changes to requirements, on budget and an agile approach. The factor ‘on budget’ was not more significant than any other factor. This factor also ranked second-last, which indicates that it is less significant than other factors for this group of respondents.

The methodology adopted traditional factor was ranked at position five and is more significant than user involvement, changes to requirements, on budget and methodology adopted agile (see Table 19). The methodology adopted agile factor ranked last and was not more significant than any other factor. This indicates that it is less critical to the success of a project. Changes to requirements did not prove to be more significant than any other factor. Changes to requirements and an agile approach are related factors, as accepting changes during the course of a project is more of an agile strategy.

4.7.3 Conclusion on project factors

The project factors were analysed and discussed above. All factors were found to be significant and critical to project success. No significant differences were noted between gender, race and roles. Project managers and software developers considered projects being on brief to be the most critical factor to project success. Project management on brief was also more significant than other factors in this study.

The factor that they considered least significant was the methodology adopted agile factor. These project managers and software developers consider this factor least critical to project success. This could be because, in this business unit, a structured approach is followed, and the team is not familiar with the agile method.

4.8 Summary of individual, team, organisational and project factors

In this section, a summary is provided on all factors and sub factors that are critical to success, from Sections 4.4, 4.5, 4.6 and 4.7. An ordered ranking of means is provided in Figure 36 that clearly shows how each factor is perceived by the respondents. The mean values were used here because these measures were normally distributed. The results in this chapter changed the conceptual framework for team and project factors. Team factors originally had 4 sub factors however resulted in 3 sub factors after the data was analysed. Project factors had 9 proposed sub factors which resulted in 12 after the study was conducted.

Figure 36 shows how each factor ranked and what the most important and least important factors were. The figure can be split into three sections i.e. top, middle and bottom, which is shown by

the dotted lines on Figure 36. All factors shown in Figure 36 are critical to project success. However, Figure 36 shows which factors are more important than others.

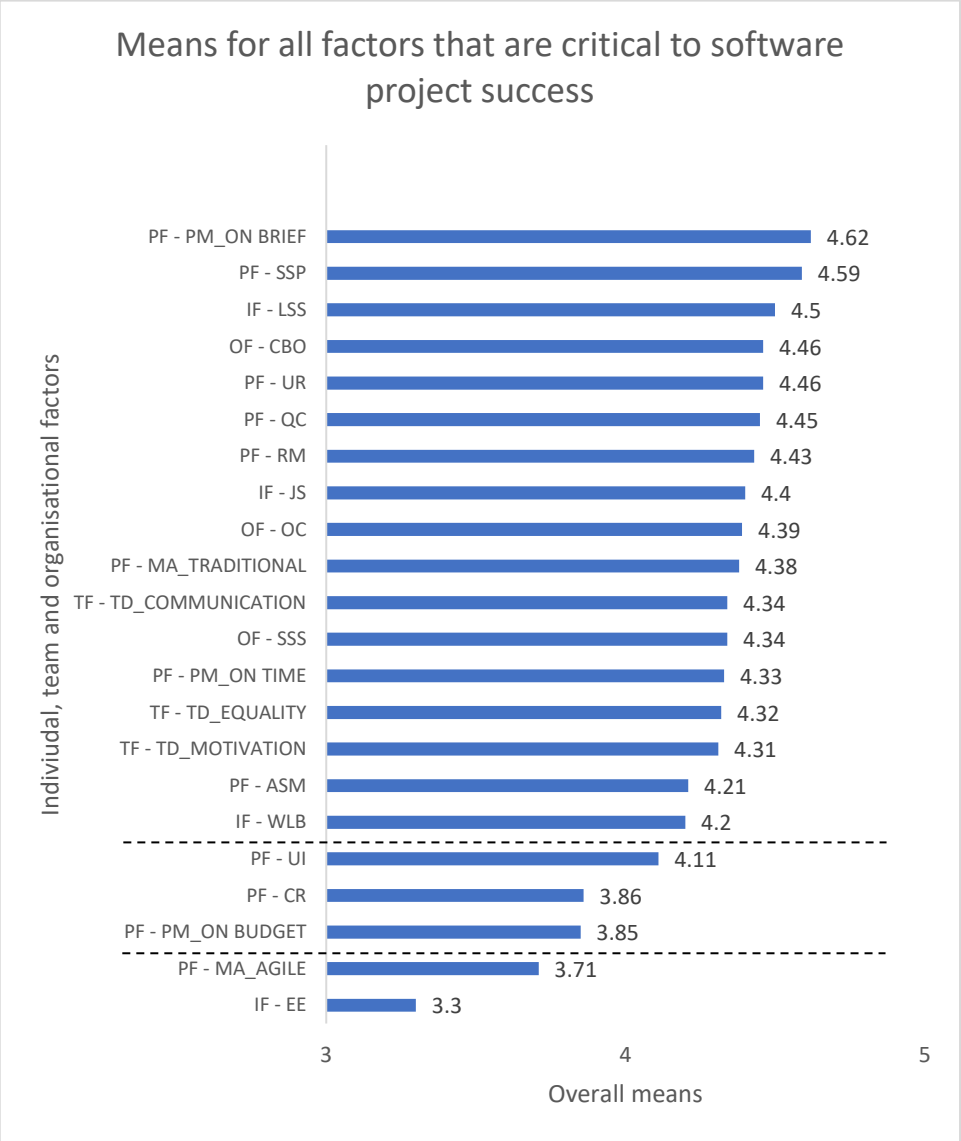

Figure 36: Means for all factors and sub factors that are critical to project success.

The factors and sub factors in the top section are more critical than those in the middle and bottom sections. The factor that was ranked first was the project factor, project management iron triangle on brief. Projects being on brief means that all requirements are met, and the project works as is expected. This was followed closely by the project factor, scope and sizing of projects. The factor is in the project management iron triangle, where it is critical for projects to be within the defined scope of the project. The scope and sizing of projects are related as both factors speak to having a definite list of requirements that needs to be fulfilled.

The level staff skills ranked third. Project managers and developers believe that staff need to be skilled in order for projects to be successful. The organisational factor, clearly defined business objectives, was fourth, followed by the team factor, understanding requirements. Considering that the three factors (on brief, understanding requirements and clearly defined business objectives) relate to requirements ranked in the top section of the graph, it shows that respondents believe that meeting and understanding requirements is critical to success.

Quality control ranked sixth and risk management ranked seventh. One could argue that if risks are identified and mitigated earlier in the project, it could improve the quality of software produced. The reverse could also be said, in that, if good quality software is produced, risks can be mitigated. Job satisfaction ranked eighth and organisational culture ranked ninth. These two factors can be related, as job satisfaction could depend on the type of environment one works in. A good organisational culture could lead to job satisfaction.

The traditional methodology adopted ranked tenth and appeared in the top section of the graph. Project managers and developers believe that following a structured approach is critical to success. Developers do, however, believe that this is more critical to success than do project managers, but not significantly so. The team dynamic factor that ranked eleventh was communication. Communication also includes accountability. Based on the interviews, participants strongly believe that communication, over the other factors, is a critical success factor. Support from management and stakeholders ranked twelfth. This factor is still in the top section of the graph, which indicates that it is an important factor for success.

The project management iron triangle factor, on time, ranked thirteenth. The team factors equality and motivation ranked fourteenth and fifteenth, with a difference of 0.01. The availability of subject matter experts ranked sixteenth on the list. This factor does not seem as critical to success as the previous ones, which could indicate that in this business unit there are enough subject matter experts. The last factor in the top section was work-life balance. This ranked seventeenth on the list and is quite surprising as software developers regarded this factor to be more important than project managers do.

The middle section consists of three factors: user involvement, on budget and changes to requirements. Whilst project managers consider having users involved in the project to be more critical to software success, developers do not. This could be because developers believe that users may add to the scope of the project. This may be why it is ranked in the middle section. Projects being on budget is more critical to project managers than developers, but not significantly so. This could be because project managers have more sight of the costing of the project than the developers do. 'Changes to requirements' is also poorly ranked, as accepting changes could cause the project to run past its deadlines. Project managers consider this factor more critical to success than do developers, but not significantly so.

The bottom section consists of two of the factors least important for success: an agile methodology being adopted and employee experience. An agile methodology is considered a less critical success factor as this business unit follows a more structured approach. Project managers do consider this factor more critical to success. Perhaps it makes it easier to manage a project that follows an agile approach. Project managers with experience have an idea of how agile approaches work, thus they would consider it. The agile approach means that software developers accept changes to requirements at any stage of the software development lifecycle. The fact that this group of respondents ranked changes to requirements and agile approaches not as critical to success, indicates that they prefer a more traditional approach. Employee experience (EE) was ranked last, indicating that even though it is a critical success factor, it is less critical than the rest of the factors in this study. The low value assigned to employee experience could be as a result of how the questions for EE3 and EE4 were phrased: 'needing more than ten years of experience for a project to be successful'.

It is interesting to note that two of the three project management iron triangle factors, on time and on brief, ranked in the top section; whereas the third factor, on budget, was ranked as the last factor in the middle section. It is also worth mentioning that all team factors ranked in the top section.

This section discussed what factors are critical to software project success. The top five factors that are critical to software project success are: projects being on brief; scope and sizing of projects; level of staff skills; clearly defined business objectives; and understanding requirements. It is also worth noting that none of the factors produced a mean value of less than three, indicating that all factors were considered critical to software project success.

4.9 Objective 5: Failure prevention

Participants were interviewed to investigate what they thought could prevent project failure. It must be noted that, when participants were asked if this business unit had a list against which project success could be measured, all participants agreed that it did not.

4.9.1 Individual factors

Participants where asked what this team does to prevent project failure. No individual factors were mentioned. When asked what this organisation does to prevent failure, 50% of the participants were not sure what the organisation does. The other 50% mentioned that they thought that having a good project manager on the project does help prevent project failure. When the participants were asked what they meant by 'good', they replied that the project manager should have experience working in this business unit. This relates to the employee experience factor. The participant was a project manager and did not have an opinion on how many years of experience were needed.

Lastly, participants were asked what they thought needed to change to prevent project failure. One of the changes proposed was to hire more technically inclined staff. This could relate to the level of staff skills factor. When asked further what they meant by technically inclined, participant three, a software developer stated:

> "We are a technical team, I don't think its fine for people with no knowledge in coding or IT to be hired. I think that the more people with a background in IT and who have studied computer science or something similar, the quicker they will understand things and it will help the project progress better" (Participant 4).

This could mean that all employees who work in the different roles in this business unit require some sort of technical knowledge (software engineering) in order to prevent projects from failing.

4.9.2 Team factors

When employees were asked what they thought the organisation does to prevent failure, the team factors that were evident were communication and accountability. This unit communicates about

all issues so that they are raised earlier, and progress update meetings are held every week. This relates to the communication factor.

A developer (participant 6) stated that everyone needs to be accountable. When asked what he meant by accountable, he responded:

> "I think everyone in the team needs to take ownership of the project. We are a team and we all want the project to be a success, but sometimes people blame others if they are wrong and that actually creates more animosity in the team, which makes it harder to work with them." (Participant 6)

4.9.3 Organisational factors

Participants did not believe that this team prevented project failure through any organisational factors. However, when asked what this organisation does to prevent failure, one respondent (a project manager) replied with having a one vision to work towards may prevent failure. This relates to the clearly defined business objectives factor.

4.9.4 Project factors

Participants mentioned that this unit breaks projects into smaller phases. This relates to the factor on the scope and sizing of projects. The participants also mentioned that proper analysis is done so that everyone understands the requirements in order to prevent failure. This relates to the 'understanding requirements' factor.

Participants were finally asked what they thought needed to change to prevent project failure. They mentioned that requirements should be defined earlier:

> "I think failure can be prevented if the requirements are defined earlier. I know we have set timelines so it makes it hard to do so, but things always change later on because dev says that it cannot be done. If the BA's knew this earlier, they could have worked around it. This would help as we won't need to do changes later on." (Participant 1)

The above relates to the ‘understanding requirements’ factor. A recommendation is that developers should be a part of the initial phases of the projects as they know what can be done from a technical perspective.

Another participant mentioned that understanding what the problem is and what they are trying to solve could help prevent project failure. This developer believed that the architects need to unpack and solve this and provide a proper solution.

4.9.4 Conclusion on failure prevention

The results from the interviews provided an understanding of what participants believed will prevent failure. However, since the population size was relatively small, only a few factors to prevent failure were identified. The individual factors that were identified were the level of staff skills (LSS) and employee experience (EE). It is interesting to note that the participant (a project manager) who mentioned that experienced employees are necessary did not state how many years of experience were necessary.

The team factors that were identified as assisting in preventing project failure were communication and accountability. The only organisational factor that participants believed could prevent failure was clearly defined business objectives. The project factors that were notable, included understanding requirements and the scope and sizing of projects.

4.10 Conclusion

The above chapter reported on the results of the responses obtained from project managers and software developers on individual, team, organisation and project factors contributing to success and failure. Many factors were found to be critical to project success. All individual factors are critical to project success. An interesting finding was that neither project managers nor software developers believed that they needed more than ten years of experience for a project to be successful. All team factors are critical to project success.

All organisational factors produced results that show that both project managers and software developers find these factors critical to project success. There was no significant difference in how the questions were answered for organisational factors. It can be said that project managers

consider support from management and stakeholders, and clearly defined business objectives, more critical to success than do developers. Developers, however, consider organisational culture more critical to project success than do project managers.

There was no significant disagreement with any of the project factors. It was found that the most critical project factor (ranked first) was project management iron triangle on brief. This means that it is crucial that the project is developed as per the required functionality. The factor that is least critical to success, according to this group of respondents, is adopting an agile methodology. This may be because the project managers and developers in this business unit work in a very structured way, indicating that they prefer a traditional approach.

Respondents were also asked to express their opinion about what factors ensure software development project success. All factors were significant. The factor that ranked first in this section was developing software of a high quality. The factor that was least significant was following a structured lifecycle. This was interesting, as this business unit follows a structured lifecycle. However, in this section they considered it to be less critical to software development success.

The respondents – project managers and developers – thought that all the factors were important enough (critical to success / will ensure software development success) that none of the composite factors produced a mean of less than the neutral value of three.

Lastly, participants were interviewed to understand what can be done to prevent a project from failing. Ensuring employees have the relevant skills and experience can prevent a project from failing. Participants also believed that understanding the project requirements is also necessary to prevent a project from failing. Communication was also a common factor. Participants also mentioned that having clear business objectives helps everyone know what they are working towards.

The next chapter discusses how the findings from this study correlate with those of other researchers, as found in the literature.

Chapter 5: Discussion

5.1 Introduction

Chapter Four presented the results from the questionnaires and interviews. In this chapter, these results are discussed in detail. Empirical evidence from the literature review (Chapter Two) assisted in providing further insights into the findings of this study.

5.2 Objective 1: Individual factors affecting software project success

The following section discusses how the results from the individual factors in the questionnaire and interviews correspond to the literature review in Chapter Two. This study investigated individual factors through several questions and the collection of demographic data. Individual factors relate to each employee on a personal level, and what constitutes their professional portfolio in terms of skills and experience.

5.2.1 Level of staff skills

The respondents from this study agreed that the level of staff skills is a critical success factor and would ensure project success. The results are similar to those of Fui-Hoon, Lee-Shang and Kuang (2001) and Tarawneh (2011), whose findings showed that constantly enhancing employee skills is beneficial to projects. Even though project managers and software developers agree that the level of staff skills is critical to software project success, project managers seem to consider this factor more important than do developers (Figure 7).

Keil, Lee and Deng (2013) investigated what skills project managers needed for projects to be successful. They found that project managers needed good communication skills in order to manage a project. This seems to agree with the views of participants in this study: when interviewed, 67% mentioned that they could improve on their communication skills. Software developers from this study also mentioned that they needed more training for their roles. This could be because technology is constantly changing; thus, the coding standards may also change.

One of the questions that was asked about the level of staff skills was if problem solving abilities were critical to project success (see LSS3 in Figure 7). The results showed that respondents agreed and strongly agreed to this question. This agrees with the views of Mtsweni, Hörne and van der Poll (2016), who found that one of the skills that developers needed to have to ensure that the project will be successful was problem solving abilities. In a study conducted by Octavianus and Mursanto (2018), it was found that skilled staff was the most critical success factor and ranked first on their list of factors.

5.2.2 Employee experience

Employee experience was investigated through a number of questions (Figure *8*). It must be noted that most respondents in this study (78%) have been working in their role for less than four years. For this group of respondents, it is not critical for project managers and software developers to have more than ten years experience for a project to be successful. These project managers and developers believe that less than four years experience in a certain role is enough for projects to be successful. These results can be compared to a study conducted by Norvig (2014), who found that to be experienced in a role, people needed more than 10 000 hours of experience. The results, however, show that having experience in the software development lifecycle, as well as working in a team, is critical to success. When views were compared, it appeared that developers considered this factor more important than project managers do.

The literature review and results found in this study seem to contradict each other. Researchers such as Alqahtani et al. (2014) and Rehman (2006) found that experience is vital for project management and software development. In their investigations, they stated that experience in a specific role offers more insight into problems faced during the development process.

Even though this group of respondents believe that more than four years of experience is not necessary for success, employees having more than ten years of experience will not be detrimental to a project either. Experience in a role is good; it means that employees are well versed in their roles and know what to expect from different types of projects. Employees with more than ten years of experience can offer guidance and mentorship, especially if a project looks like it is going to fail.

5.2.3 Work-life balance

In the work-life balance section, three questions were posed to project managers and developers (see Figure *11*). Working after hours and on weekends was not statistically significant. It was also excluded from the composite mean value of work-life balance.

Project managers' and software developers' perceptions about work-life balance agree with a study conducted by Perera (2011). Perera found that employees who have a good work-life balance usually perform optimally at their jobs. It can be deduced, from the results that having a good work-life balance and not working overtime or on weekends is critical to success. This means that employees who have enough time to focus on all aspects of their lives work better, which further contributes to project success. Arif and Farooqi (2014) also found that employees who have a good balance between work and their personal lives, tend to show a greater commitment to their organisation.

Whilst the developers and project managers in this study consider this factor critical to project success, it must be noted that both groups of employees work overtime and on weekends. Developers seem to work more overtime, and on more weekends, than do project managers. The developers that work overtime are usually team leaders or are working on more complex projects. In this study work-life balance was the first of two success factors where project managers and developers answered significantly differently.

Heeks et al. (2017) found that work-life balance was an important project success factor. They saw that employees needed to have a good balance so that they are not stressed, as stressed employees may produce poor quality work (Heeks et al., 2017). Whilst project managers' and developers' views were more spread out for this factor, some of them do believe that ensuring a good balance between time at work and personal time is critical to project success.

5.2.4 Job satisfaction

The results in Chapter Four show that almost all the respondents agree that job satisfaction is critical to project success (see Figure *12*). When the views of project managers and software

developers were compared, it was found that they held similar beliefs on this factor. Enjoying the work that one is given is a major factor of success, as then employees do not tend to see their jobs as hard tedious work, but rather as an enjoyable task that gives them a sense of accomplishment when done. Linberg (1999) also found that developers considered a project to be successful if they were satisfied with their jobs, and could learn from the project and be creative (see Section 2.6.5).

The study conducted by Halkos and Bousinakis (2010) stated that increased job satisfaction leads to increased productivity. This can be seen in the results of this study, as project managers and software developers regard job satisfaction as a valuable success factor. Halkos and Bousinakis found that satisfaction levels are not necessarily based on remuneration. It must be noted that, in this study, developers who work longer hours and overtime receive remuneration for their efforts; whereas project managers do not. This factor, job satisfaction, was the other success factor on which project managers and developers differed significantly.

Work-life balance and job satisfaction seem to go hand in hand. This means that if employees do not have a balance between their personal time and work, they may not be satisfied in their jobs. The results in this study showed that developers worked more overtime, and on more weekends, than did project managers, and it therefore made sense that developers considered these two factors more important than project managers did. This may be because they do not have a good work-life balance; therefore, it is more important to them.

5.3 Objective 2: Team factors affecting software project success

The following section discusses whether the results from the questionnaire and interviews on team factors correspond to the literature reviewed in Chapter Two.

5.3.1 Communication

The communication factor was the first team dynamic factor that was investigated. Respondents believe that good communication between team members is critical to project success (see Figure *15*). It is necessary for team members to know about everyone's projects. Communicating in person and communicating regularly will ensure project success. Communication also ranked in

the top section of Figure 36. This study agrees with those of Estler et al. (2014) and Sheffield and Lemétayer (2013), which identified communicating as a guiding factor for a project to be successful. Project managers considered the questions under the communication factor as more critical to success than developers did. This could be because project managers are responsible for communicating the status of the project and facilitating communication between the different people involved.

5.3.2 Being treated fairly

Both groups of employees, project managers and software developers, agree that being treated fairly is critical to project success. This factor was ranked in the top section of Figure 36. Beecham (2014) found that when employees are treated well, they feel valuable and they produce work of good quality, which may lead to project success. In this study, project managers considered this factor more critical to success than developers.

5.3.3 Accountability

Accountability was investigated through three questions. Both project managers and developers believe that being accountable is critical to project success (see Figure *17*). McHugh, Conboy and Lang (2011) stated that accountability promotes work ethic which in turn ensures that all employees work well together. Reed and Knight (2010) also found that everyone in the team being accountable for their own work, their project and role, will promote good project performance. The results from this study agree with that of McHugh et al. (2011) and Reed and Knight (2010). In this study, on further analysis, accountability was combined with the communication factor.

5.3.4 Motivation

Motivation was the last team factor that was investigated. The results from this study are similar to those of Takpuie and Tanner (2016), as they found that team members may be motivated by financial rewards, enjoyment of their job and peer recognition. It is interesting to note that developers consider this factor as more critical to success than do project managers. It must also be noted that developers are rewarded for their overtime efforts; whereas project managers are

not. This may be why developers seem to think that motivation through remuneration is critical to project success.

5.4 Objective 3: Organisational factors affecting software project success

The following section discusses whether the results on organisational factors correspond to the literature reviewed in Chapter Two. All organisational factors were considered critical to project success. Clearly defined business objectives appeared in the top five factors of all the ranking lists, showing that it is critical to success.

5.4.1 Support from management and stakeholders

Figure *21* shows the responses to questions whether support from management and stakeholders is critical for software project success. As can be seen from the results, project managers and software developers share similar views on this factor. The responses show that both groups of employees agree that management and stakeholders being involved in the project by offering solutions and participating in meetings does contribute to project success. This factor ranked in the top section of Figure 36.

Dezdar and Ainin (2011) found that support from management and stakeholders positively affects project success, which agrees with the results of this study. Hastie and Wojewoda (2015) also found that managers and stakeholders providing emotional and financial support will ensure the success of a project. This will encourage employees to do their jobs better which will increase employee productivity. Project managers in this study consider this factor more critical to project success than developers do. This may be because project managers are required to interact with stakeholders and management more than developers do. Octavianus and Mursanto (2018) ranked this factor as fourth in their study. This study shows similarities to that of Octavianus and Mursanto (2018), as this factor ranked in the top section of Figure 36.

5.4.2 Clearly defined business objectives

Based on the results in Figure *22*, it can be deduced that project managers and software developers both believe that clearly defined business objectives for each project are critical to project success. Most of the results showed that respondents either agreed or strongly agreed with the variables under this factor. The respondents also believe that business objectives provide a sense of direction and purpose and motivate the team to work towards a common goal. Project managers consider this factor more critical to project success than do developers. This factor also ranked in the top section of Figure 36.

When respondents were asked whether clearly defined business objectives ensured software development success, all project managers strongly agreed with this statement. This factor also ranked in the top section of Figure 36. Hastie and Wojewoda (2015) found that business objectives should align to the organisation's goals and strategy; thus, having them clearly defined is critical to the success of a project. This agrees with the current study. Tarawneh (2011) also found that if business objectives are well defined, everyone in the team will understand the goal of the project and this may ensure project success.

5.4.3 Organisational culture

Project managers and software developers agree with most of the questions that make up the organisational culture factor. Respondents believe that a friendly and supportive working environment is critical to project success. Respecting others and showing empathy also promotes good organisational culture. Leadership setting examples by following company values, rules and procedures also is critical to success, as employees will have standards to follow (see Figure *23*).

Gu et al. (2014) showed that organisational culture affects how employees perform in their jobs, thus having a direct impact on project success. Tarawneh (2011) also found that how employees perform is based on the organisational culture. It is therefore necessary to ensure that there is a good organisational culture for employees to work in, so that they may work optimally, which will ensure project success. This study therefore shows similarities to those of Gu et al., (2014) and Tarawneh (2011) in that organisational culture does affect project success. Robinson and

Sharp (2005) also investigated what impact organisation culture has on project performance in a multinational bank. The organisational culture was strict, where employees were told what to wear, how they worked and how they interacted. It was found that, even though the projects were completed on time, the employees were not satisfied with the working environment. This shows how important organisational culture is to project success.

The fact that in this study, project managers and software developers believe that organisational culture leads to project success could relate to their belief that team motivation, accountability and equality also leads to project success. The team dynamics factors relate closely to organisational culture in that it sets the tone for how team members collaborate and work together.

Tornjanski et al. (2015) remarked about how banks were originally regarded as a conservative industry who had a slower pace of doing anything innovative. The fact that this unit uses a traditional approach to software development could support that view. Comparing that view to the bank in this study, where it was voted the most innovative bank, the perception that all banks are conservative may be misguided. The assumption that banks are conservative may also be mistaken in the dress code of employees. In this business unit in this bank, employees dress casually, and no strict dress code is enforced. This, however, cannot be generalised to all other units within the bank. In this unit, working hours are also flexible as long as they are agreed upon with managers so employees can start and end at any time, given that they work their working hours. These are factors that contribute to organisational culture and it is important for all employees to understand and be aware of these (Pathak, 2018).

Khoza and Marnewick (2020) also discuss management styles. These included command and control as well as leadership and collaboration. In this unit a command and control management style is followed due to the hierarchical reporting structure within the organisation. The management style also affects the organisational culture as employees on each level of the hierarchy are treated and managed differently and have different work assignments. This may further tie in with job satisfaction and work life balance factor. The organisational culture sets out how the working environment will be. Based on this, the environment can either create a positive or negative experience for employees which may further affect their hours worked and even their satisfaction with their job. Based on the above discussions, this organisational and company culture appears to be flexible, as employees have the freedom with dress codes and working hours but stricter when it comes to getting the work done and meeting deadlines.

5.5 Objective 4: Project factors affecting software project success

The following section discusses whether the results on project factors correspond to the literature reviewed in Chapter Two.

5.5.1 Understanding requirements

Figure 26 shows that respondents mostly agree that understanding requirements is critical to project success. The overall mean for the 'understanding requirements' factor ranked in the top section of Figure 36 which also shows that this was considered one of the most critical factors in the study. It was ranked 5th. Project managers consider this factor more critical to project success than do developers but not significantly so.

The results from this study show similarities with those of Poon and Wagner (2001) and Hussain and Mkpojiogu (2016). They stated that understanding the requirements of the project is vital to project success. The results from their study showed that having sessions and meeting with users and stakeholders to understand the requirements plays a major role in understanding and documenting requirements.

Hussain and Mkpojiogu (2016) mentioned that the requirements engineering phase was the foundation of all projects. Thus, it is necessary to find a good technique to gather and document the requirements. Good documentation will provide a clear guide to what is required from the project (Hussain & Mkpojiogu, 2016). It is interesting to note that employees in the business unit in this study use methods of documentation such as functional and system requirements specifications as well as screen flows. Based on the results, it can be deduced that understanding the requirements stated in the functional and systems requirements specifications, as well as the screen flows, is critical to project success. It can also be noted that having regular sessions to unpack requirements also contributes to project success.

5.5.2 Changes to requirements

This study investigated where accepting changes to requirements was critical to project success. Even though the standard deviations were high for each of the questions, which also showed a spread of responses in Figure *27*, most respondents believed that accepting changes is a critical success factor. Project managers seemed to consider this factor more critical than developers. This may due to the fact that changing requirements may require more coding or changing existing code, which can appear tedious to developers. The composite factor was not more significant than any other project factors.

Changes to requirements is typically associated with agile approaches; however, it is interesting to note that this business unit follows a traditional approach. This could be why this factor was considered less significant than the others. Project managers and developers in this study could be less accepting of changing requirements because it requires more work, time and effort.

Hastie and Wojewoda (2015) and Serrador and Pinto (2015) stated that changes can occur at any stage during projects. Thus, it is important to have a good change management process in place to manage this. They also mentioned that using an agile approach would help to accommodate changes much more easily.

Uskov (2017) and Gravell et al., (2017) stated that accepting changes will be beneficial to a project. However, the correct methodology, agile, must be followed so that these changes are easily accepted. It is interesting to note that, even though this team does not follow an agile approach, they still accept changes for projects and consider this to be critical to success. It must be mentioned that respondents may feel as though changes that come from users, stakeholders or subject matter experts may contain enhancements that will improve the usability of the functionality. Changes from subject matter experts could be a change in the architecture of the software and changes from stakeholders could be items that will ensure business benefit. Even though the results from this study show that accepting changes is critical to project success, this factor should be more significant if an agile approach was followed.

5.5.3 Quality of software

The quality of the software produced is very important as customers will be using the functionality and it needs to work as per the specification. Project managers and software developers mostly agreed with the statements about the quality of software, as can be seen in Figure 28. Quality control also ranked in the top section of Figure 36. This shows its significance in this study.

Wasserman (2010) indicated that testing the software on devices that customers will be using is vital to ensure the validity of the software developed. One of the questions posed in this study was whether respondents believed that the software working on all platforms (android, iOS and mobi) is critical to project success. The results from this study align with that belief, as respondents agreed that the functionality must work on all platforms.

Project managers and developers believe that the quality control is a critical success factor. It must be noted that the questions that were asked in the questionnaire are basic tests for each software project that project managers and developers work on. These are the first tests conducted when testing a project to ensure that the quality is good, and that it works as per the specification. Attarzadeh and Ow (2008) stated that, to ensure the good quality of software, precautions must be taken to prevent and remove software defects. When testers begin testing projects, they must be able to log defects so that developers can fix them before the projects are released to users.

In a study that investigated what developers thought software project success means, it was found that the quality of software produced was critical to project success (Linberg, 1999). This shows alignment with this study, and both groups of respondents consider this factor a critical one.

5.5.4 Availability of subject matter experts

Subject matter experts in this business unit include solutions and software architects that usually have experience with the product being developed and can offer insights into the best way to develop the project. Based on the results, project managers and developers agree that having subject matter experts available during the project is critical to software success. Project managers considered this factor more critical than developers did. In the composite measurements, it can

be seen that ASM1 (sufficient subject matter experts (solution/software architects etc.) in the project team) was excluded from the resultant mean. In Figure 36, the availability of subject matter experts ranked in the top section. However, when respondents were asked if availability of subject matter experts would ensure software development success.

McIntosh, Kamei, Adams and Hassan (2016) stated that having an abundance of subject matter experts does ensure the success of a project. This is shown in the results of this study, as respondents agree that sufficient subject matter experts contribute to success. Respondents also agree that not having to wait for subject matter experts' input, as well as having them readily available when needed, ensures project success. Castillo et al. (2016) also found that subject matter experts are necessary for project success as they have knowledge of the systems, design the solution and assess the performance of the system.

5.5.5 Scope and sizing of projects

Project managers and developers were asked whether they thought the scope and size of a project is critical to project success. The scope and size of projects are usually determined by the subject matter experts and this guides the project requirements and timelines. Respondents agreed that this factor was critical to success. Respondents were asked if having projects with smaller scope and size was critical to project success, and most respondents were neutral towards the statement. The standard deviation resulted in a value of 1, indicating that responses were spread out.

The composite mean value of 4.59 showed that this factor was considered a critical success factor. However, SSP3 (projects with smaller scope and size), SSP4 (projects that are broken into smaller phases) and SSP5 (projects that can be completed within a realistic/practical time) were excluded from the value. Developers mostly considered this factor more critical than do project managers. This may be because, when projects that are broken into phases, it means less work than combining all phases into one large project. This factor was more significant than eight other team factors and ranked second on the list of most critical success factors. Respondents were also asked if well-sized projects will ensure the success of a software development project, and they agreed. For Octavianus and Mursanto (2018), project size also ranked in the middle.

Tarawneh (2011) stated that minimising the project scope contributes to project success. Whilst responses are spread out, developers believe more strongly that breaking projects into smaller phases is critical to project success. The overall results from the factor showed that both groups

agree that accurately sized and scoped projects are critical to project success. Thakurta (2013) also found that, by increasing the project scope, it puts pressure on the team, which can have a bad impact on the project. This study agrees with the current study, as respondents believe that projects with smaller scope and size may be more successful. In a study conducted by Keil et al. (1998), they found that the size of a project is dependent on the scope; thus these two factors are closely related.

5.5.6 Project management iron triangle

The project management iron triangle was split into three sub-factors: on brief, on time and on budget. The on brief factor was ranked first out of all team factors, indicating that project managers and developers believe that this is the most critical success factor. Projects being 'on brief' mean that the functionality delivered must be what was initially specified. It is interesting to note that, when the ranking was conducted to see which team factors were most important (see Figure 36), the scope and sizing of projects was ranked second. Understanding requirements is also another factor that relates to projects being on brief and this was also ranked higher up on the list. It can be deduced that this factor is seen as a critical success factor by project managers and developers.

The results from this study show similarities to those of Kapczyński (2016), where it was found that one of the most critical success factors was the that the project needed to comply with the project management iron triangle. Kapczyński's study considered this as one factor, even though it contained three sub-factors.

Another study, conducted by Neves, Borgman and Heier (2016), showed that their respondents believed that, for the project performance to be good, the project team needed to ensure that the project was on schedule and within budget. This differs a bit from the results produced by project managers and developers in this study. Even though all project management iron triangle factors implied successful software development projects, on brief and on time were always more critical than projects being on budget.

In a study done by Lech (2013), who investigated what factors affected project success, it was found that projects being on budget was crucial to project success. Again, this differs to this

study, as even though it is a critical success factor, it was considered the least critical in this study by this group of respondents. The results from Linberg (199) contradict this. Developers thought that a project was successful when they had delivered the required functionality, even if the project was late and over budget.

Wateridge (1998) investigated what project managers believed was critical for projects to be successful. It was found that meeting user requirements was one of the main factors. This is similar to what respondents in this study believed, as projects being on brief, meeting the requirements, was most critical to success.

5.5.7 Applying risk management

Project managers and software developers are in agreement that risk management is vital to any project. Identifying risks at the beginning of the project, taking precautions to prevent the risks and continuously monitoring the risks will ensure that the project is successful. There was no disagreement with any of the questions for this factor. It was also ranked in the top section of Figure 36.

Aloini et al. (2007) stated that the main reason why projects fail is that risks were not identified and mitigated. This study agrees with Aloini et al. (2007) in that the project team needs to ensure that the risks are mitigated as they are identified. De Bakker et al. (2010) stated that, in order for projects to be successful, risks need to be identified and ordered by priority, and then a solution needs to be designed to mitigate the risks.

5.5.8 Methodology adopted

The 'methodology adopted' factor investigated whether respondents prefer a traditional or agile approach. The questions related to traditional approaches produced results that indicate that project managers and developers consider this factor is critical to project success. The results also showed that developers find the traditional methodology more critical to success than do project managers. The traditional approach was considered more significant than the agile approach and ranked in the top section of Figure 36. It is interesting to note that, when

respondents were asked if following a structured lifecycle would ensure software development success, responses were spread out.

When the participants considered adopting an agile approach, it was not considered as critical as the traditional approach. The standard deviations were high for the questions relating to the agile approach, indicating that the responses were more spread out. Even though the results also show that agile approaches are less critical to success, project managers consider an agile approach more critical than developers do.

Theocharis et al. (2015) showed how a hybrid approach offered more benefits than a single methodology. The structured traditional approach is used so that every team member knows what they are responsible for. However, during development and testing, limitations could arise that cause the team to adopt an agile attitude to ensure the project is implemented and the best solution is delivered.

This organisation takes six months to implement a software development project (see the Background, section 1.3). As indicated in the background, one phase must be completed in order for the next to begin. This shows that this business unit follows a waterfall traditional approach. Khoza and Marnewick (2020) however found that the average traditional software development project takes between 15-18 months, whilst agile projects take approximately 10 months to complete in South Africa. Comparing these timeframes with those of this study, it can be seen that whilst this unit follows a traditional process, it has much shorter timelines indicating that a mixed software development approach is easily accepted and adopted by this business unit.

The adoption of both traditional and agile practices can be seen from the responses from the questionnaire. An agile methodology is more accepting of changes to requirements; thus, this factor can be compared to the changes to requirements factor. It was found that both project managers' and developers' responses ranged from 'strongly disagree' to 'strongly agree' and are similar to the responses to methodology adopted. The organisational culture also promotes a flexible working environment where employees are not governed by a dress code or strict working hours. The business unit is thus obtaining the flexibility of the shorter project length, with the structure of the traditional approach. This offers insight into why they can readily adopt both traditional and agile practices.

5.5.9 User involvement

Project managers and developers from this study agree that users being involved in the project is critical to success. Both groups of employees agree that users should be available to work with project managers and developers. However, project managers consider this factor more critical than developers do. Developers may think that users may add more requirements to the project, thus causing them to rework and add new code to accommodate the changes from the users. This may be why they do not consider this as critical as project managers do. This factor ranked in the middle section of Figure 36, which shows that while this factor is important to this group of respondents, there are other factors that are much more important. Software is usually developed for users, hence user involvement is seen to be necessary, as users provide the underlining requirements and they could possibly test the prototypes.

This study shows similarities to those of Bano and Zowghi (2015) and Tarawneh, (2011). They found that a lot of user involvement ensures that a project will be successful as they will be able to test the software and provide feedback on what can be improved. Octavianus and Mursanto (2018) ranked factors from most to least important. User involvement ranked third on their list, showing that their group of respondents believed that it is quite critical to project success. In this study, however, this factor was not considered as important as other factors.

5.6 Objective 5: Failure prevention

5.6.1 Individual factors

When respondents were interviewed and asked what they thought would prevent project failure, two individual factors surfaced: the level of staff skills and employee experience. Participants believed that one of the ways failure can be prevented was by hiring more people with a technical background. This is because they work in a technical environment and it is beneficial to hire employees with some sort of experience in working in this type of environment. This result is similar to a study conducted by Stellman and Greene (2005). They found that it was necessary

to ensure that the company hired staff with the appropriate skills for their roles and that the company utilises those skills to the best of their ability. Doing this will prevent project failure.

Participants also mentioned that having a good project manager on the project will help prevent the project from failing. Participants believe that a good project manager is someone with experience in the field.

5.6.2 Team factors

The team factor that was mentioned was communication. It is necessary for the team to communicate about the progress of the project, as well as if there are issues so that they can be sorted out earlier on in the project.

5.6.3 Organisational factors

The responses for organisational factors were limited and the only factor that participants were confident would prevent project failure is having clearly defined business objectives. This aligns with a study done by Stellman and Greene (2005), who found that clearly defining business objectives drives employees to work towards the company's visions and goals.

5.6.4 Project factors

A few project factors were mentioned when participants were interviewed about how to prevent project failure. Participants believe that breaking projects into phases can be used as a failure prevention technique, as when projects are oversized or over-scoped, the work load becomes unrealistic. This agrees with the views of Kaur and Sengupta (2013). They found that if projects are accurately scoped and sized, it will prevent them from failing.

Another project factor that project managers and developers believe will help prevent projects from failing is understanding requirements. They mentioned that the requirements need to be

analysed properly and documented so that there is always a point of reference. This also emerged in the study conducted by Kaur and Sengupta (2013). They found that if requirements are documented clearly, it will help the team understand what needs to be done and there will be less uncertainty in the team.

5.7 Conclusion

This chapter showed the relationship between the research questions, the interview questions and the literature review in Chapter Two. This chapter discussed what factors were critical to project success and what can be done to prevent project failure. It is evident from the literature review and the empirical research that there are multiple factors that are critical to project success.

The individual factors included the level of staff skills, employee experience, work-life balance and job satisfaction. The results obtained in this study for individual factors were very similar to those in previous research. One of the only factors in the study that differed from previous research was the employee experience factor. This study asked respondents if more than ten years' experience was critical to project success, and respondents disagreed with this statement. Respondents in this study did not believe that employees needed more than ten years' experience for projects to be successful; perhaps because most employees in this business unit had less than four years of experience. Employee experience also ranked last when critical success factors were ranked. It is also worth noting that project managers and developers answered questions about work-life balance and job satisfaction significantly differently.

All organisational factors were considered to be critical success factors. It is also interesting to note that all organisational factors ranked in the top section of the ranked list. This shows that support from management and stakeholders, clearly defined business objectives and organisational culture have a direct impact on project success.

All project factors were considered critical to project success. The most critical factor, as seen by respondents in this study, was the project management iron triangle – on brief. This factor showed that if a project is on brief and developed as per the requirements, the project will be successful. Many researchers found that the project management iron triangle was the traditional way of measuring a project's success. However, over the years, more and more factors were added to various models to assist the IT industry in measuring software project success.

A project factor that produced conflicting results is ‘changes to requirements’. The results show that respondents in this study believe that accepting changes is critical to project success. However, many researchers found that accepting changes is usually related to following an agile lifecycle, which this business unit does not do. It is also interesting to note that when respondents were asked about which lifecycle would ensure project success, following a structured approach was ranked last. This business unit follows a structured approach, but it appears as though respondents may prefer a mixed approach.

Lastly, techniques to prevent failure were investigated. Based on the results from the interviews and the literature, several techniques were identified. These included hiring more technically inclined people; making sure requirements are well documented; good team communication; that projects are scoped and sized accurately; and that everyone is accountable for their work.

The next chapter concludes the study.

Chapter 6: Conclusion

6.1 Introduction

This research investigated project managers' and software developers' perceptions of software development project success. The research further identified the factors that are critical to project success overall and what can be done to prevent project failure.

A literature review was also conducted to determine what factors have been identified as contributing to project success in previous studies. The literature review provided an understanding of how different software development projects are conducted and what project managers and software developers believe to be success factors. The research study followed a mixed design approach, collecting data through questionnaires and interviews. Project managers and software developers from a digital banking business unit in a South African bank answered these questionnaires and attended interviews. The questionnaires were completed by 41 participants – six project managers and 35 software developers. Six interviews were conducted with two project managers and four software developers to gain further insight into their perceptions. This study had a 100% response rate to the questionnaires.

6.2 Conclusions

6.2.1 Individual factors

Four individual factors were investigated to understand how individual factors affect project success. The first individual factor was the level of staff skills. All questions for this factor were significant, indicating that the level of staff skills is a critical success factor. It is evident from this study that project managers need to have the necessary management skills to be able to effectively accomplish the goals of the project and software developers need the relevant coding skills to develop a project.

Employee experience was investigated next. It was found that there was a disagreement on two of the four questions. Project managers and developers do not believe that employees with more than ten years of coding experience and project management experience is critical to success. It can be deduced that, for these respondents, more than ten years' experience is not directly related to project success. Experience in the same job, but in a different business unit, might not always add value to employees as different teams and units work differently. This factor showed that even though two of the questions were outliers, employees still believe that experience in working in a team, as well as with the software development lifecycle, is critical to project success. Experience in working with a team helps employees develop team and people skills. Experience with working with the same software development lifecycle helps employees understand and know what is expected from each team member and themselves.

When employees were asked about work-life balance, they did not believe that working after hours or on weekends contributed to project success. It is important that a good balance exists between the working and home environments, and that employees are happy in both environments, to promote productivity. An increase in productivity implies that employees work better, which is critical to project success. Work-life balance and job satisfaction are related to each other. Employees believe that being satisfied in their jobs will ensure project success. Job satisfaction stems from enjoying the work that is assigned to one, being appreciated for the work that one does, having good relationships and working environment, and successfully solving problems. All employees agree that these factors are critical to project success. The only two critical success factors in this study where project managers and developers responded significantly differently, were for work-life balance and job satisfaction. In both factors, developers responded with more agreement to these factors than project managers did.

6.2.2 Team factors

Communication, being treated fairly, accountability and motivation are critical to project success. Communication is a critical success factor, according to both project managers and developers. When the results were analysed, the accountability factor was combined with the communication factor. Respondents also believe that being treated fairly is a factor contributing to success. Being given the same work as everyone else, as well as work being allocated in the same way, allows team members to believe that they are treated the same and reduces animosity in the team, which promotes productivity. Motivation is also critical to success. Receiving recognition and financial

rewards do contribute to motivation, which drives employees to work better, thus contributing to project success.

6.2.3 Organisational factors

Three organisational factors were investigated to understand whether they impacted on project success. The first factor was support from management and stakeholders. It is necessary for managers and stakeholders to be involved in all project-related meetings as it shows that they are interested and committed to the project. Employees feel that projects will be successful if managers and stakeholders are involved. Project managers and software developers feel that having managers or stakeholders define a solution is critical to success as they have the main business objective in mind.

Clearly defined business objectives was the next factor that was investigated. Project managers and developers agree that having objectives clearly defined is critical to project success, as there is a clear goal and vision to work towards. The last factor was organisational culture. When employees were asked questions on their working environment, they agreed that a friendly, supportive, quiet working environment does contribute to project success. Employees believe that most of their time is spent at work so there is a need for their working environment to be comfortable. Empathy is also very important to ensure that people are treated well. Employees in leadership roles need to set a good example by following company values. Project managers and software developers believe organisational culture is critical to success as employees spend most of their time at work; therefore the environment and culture needs to promote a good atmosphere, so that employees can be productive.

6.2.4 Project factors

All project factors are critical to project success. The first critical factor is understanding requirements. It is necessary for the project team to understand what is required for a project so that the correct functionality is developed. The results showed that there was a significant agreement with all variables under this factor. ‘Changes to requirements’ was another factor that was investigated. It is interesting to note that employees believe that accepting changes during the software development lifecycle does, in fact, contribute to success. Employees want to put the best solution forward; thus, accepting changes that will improve the solution ensures that the project will be successful.

The quality of software produced is a factor that is critical to success. The quality usually determines how users will interact with the software produced. The customer experience must always be kept in mind when building software, as they are the end users. The results show that employees agree that this factor is critical to project success as it affects the product that is being produced. Availability of subject matter experts is also a critical success factor. Subject matter experts in this business unit refer to software and solution architects. Project managers and software developers agreed that subject matter experts must be available during projects to help and provide solutions. Their availability during the project and their input is vital to a project success.

The scope and size of a project defines how the software will work and how many resources need to be assigned. It is worth noting that the question that asked if having projects with a smaller scope and size contributes to success, resulted in mostly neutral responses. This factor also ranked second on the ranking list for critical success factors, indicating its importance to this group of respondents.

The project management iron triangle was investigated and split into three factors: projects being on time, on brief and on budget. The most critical factor in this study proved to be projects being on brief. This factor relates to the scope and sizing of projects as well as understanding requirements. The project being on budget was ranked last, indicating that this is the least critical success factor for these respondents.

Risk management was the next team factor that was investigated. It was deduced that risks need to be identified and mitigated so that customers and end users are protected. Respondents were asked whether a traditional or agile methodology was critical to software project success. The results show that respondents prefer both approaches. However, project managers seem to align more with agile approaches, whilst developers align with traditional approaches; but not significantly so. This team may prefer a hybrid approach.

User involvement was the last team factor that was investigated. Having users involved from the start of a project to help define requirements and test the prototypes helps to pick up issues earlier and rectify them. Employees are in agreement that this factor was critical to project success.

6.2.5 Failure prevention

Project managers and developers were interviewed to understand what they thought would prevent a project from failing. A few techniques were found. These included hiring more technically inclined people; ensuring requirements were well documented; ensuring that there is good communication between team members; being accountable and having clear business objectives.

6.3 Overall view of factors

This study investigated what factors contributed to the success and failure of software development projects from a project manager and developer perspective. It became clear that many factors were seen to be critical to project success. The top five factors on the ranking list were a combination of individual, team, organisational and project factors. This shows that project success is multi-faceted.

The respondents in this study agreed that all factors were critical to success as the means were all greater than three. This shows that all factors were considered important. This means that multiple factors should be used to measure the success or failure of a project. It is expected that factors found to be important in this study (understanding requirements, scope and sizing of projects, delivering the required functionality, level of staff skills, clear business objectives) would be important to other teams as well – although this needs to be confirmed by future research.

The success factors that project managers and developers answered significantly differently on were work-life balance and job satisfaction. Developers considered work-life balance and job satisfaction more critical to project success than did project managers. These results may have been produced due to the fact that developers work longer hours and, on more weekends, than do project managers, which means that developers hardly have time to do things in their personal time. It was surprising that project managers and software developers agreed on almost all factors as these two groups of individuals have very different backgrounds and drive the project from different perspectives.

Research in the existing literature showed that there are multiple factors that contribute to the success of projects. These factors differ based on different environments and industries. After thorough investigation, there appeared to be few studies on the factors of software development project success in the banking industry. This study attempted to fill that gap by investigating the perceptions of project managers and developers in a South African bank's digital banking business unit. Software development teams which have similar environments to this banking environment could find that they have similar sets of important success factors – although this would have to be confirmed by further research.

6.4 Limitations of the study

The results of the study cannot be generalised to all business units within this South African bank. They can also not be generalised to other banks in the country. However, it is expected that software development teams with similar team structures and environments to this banking software development team would have similar results.

The views of stakeholders, sponsors, business analysts, system analysts, managers and testers were not included in this research. This study therefore only offers the views of two groups of respondents who work together and cannot be generalised to the other groups. This study also only focused on a single bank, offering limited insight into the South African banking industry. Other banks in South Africa also develop software solutions for their customers to utilise. The results can also not be generalised to these other banks.

Only internal factors were investigated to understand what factors affected project success in this business unit. External factors were not included in this study.

6.5 Recommendations to ensure project success

Based on the results from this study, a few recommendations follow:

- The project requirements must be well defined and documented so that all team members have a guide to follow. This is also necessary so that the correct requirements are developed and tested.
- The business unit must ensure that staff continually gets the proper training to obtain the necessary skills to do the job.
- It is important to hire more technically inclined employees with experience working in a technical environment.
- All the project team members must communicate about the project to ensure that everyone knows what is going on and any issues that are raised can be fixed sooner.
- All team members must be accountable for their work and management needs to act when employees are not accountable.
- The team needs to have clearly defined business objectives so that everyone knows what they are working towards and have a common vision in mind.
- The software and solution architects need to ensure that projects are accurately scoped and sized.
- Management must ensure that their employees have a good work-life balance and are satisfied in their jobs.

6.6 Recommendations for further research

The research could be expanded in several ways:

- This study only aimed to understand the perceptions of project managers and software developers. However, there are many other roles within the project team, and these team members' perceptions could also be explored. They include business analysts, systems analysts, designers and testers.
- The study could be expanded to other business units within the South African bank.
- The study could be expanded to other banks within the country.
- The conceptual model could be tested in other software development environments, and in other roles in the software development team.

Success is important to all projects as a lot of time and resources are spent on developing a product that adds value to users. Success means different things to different people and will always be

argued by researchers as they attempt to define it. It is important to define what success means to a specific person, business unit or company, so that a standard set of success criteria can be created, and projects can be assessed. This study showed that factors critical to software development project success are perceived very similarly by software developers and project managers, with the exception of job satisfaction and work-life balance, which are more important to software developers. The most important factors that lead to success are also in the list of important factors that cause software development failure. This study suggests that, if one could ensure that projects are on brief, accurately scoped and sized; that employees have the appropriate skills for the job; that the business objectives are clearly defined and are achieved; that requirements are understood; and that good quality software is produced, then the project will be successful in this software development context.

Appendix A: Questionnaire

University of KwaZulu-Natal

Researcher: Aruna Seerpath / 0746481598 / aruna.seerpath@gmail.com

Supervisor: Sue Price / 031 260 3162 / pricec@ukzn.ac.za

Project Title: Project managers' and Developers' Perspectives of Software Development Project Success within a South African Bank

This study has been ethically reviewed and approved by the UKZN Humanities and Social Sciences Research Ethics Committee (approval number_____).

Date: _____

Consent form

Dear colleague,

I am an MCom (IS&IT) student in the School of Management, IT and Governance. Please would you consider participating in my research study which is entitled "Project managers' and Developers' Perspectives of Software Development Project Success within a South African Bank"?

It aims to investigate software development project success from a project managers' and software developers' perspectives within a South African bank. The study is expected to include 30 questionnaire participants and 10 interviewees.

Completing the questionnaire will take around 15 minutes.

- ✓ The information you provide will be used to conduct an analysis on software development project success from project managers' and software developers' perspectives.
- ✓ Your name will not be written on the questionnaire or be kept in any other records. We kindly ask you to fill it out with as much accuracy as possible.
- ✓ All responses you provide for this study will remain confidential. When the results of the study are reported, you will not be identified by name or any other information that could be used to infer your identity. Only researchers will have access to view any data collected during this research, and the responses will not be used for any purposes outside of this study.

- ✓ Your participation is voluntary and you may withdraw from this research any time you wish.
- ✓ Your refusal to participate will not result in any penalty or loss of benefits to which you are otherwise entitled to.
- ✓ All data, both electronic and hard copy will be securely stored during the study and archived for 5 years. After this time, all data will be destroyed.
- ✓ The research intends to abide by all commonly acknowledged ethical codes. You will receive no incentive or payment for your participation.

If you have any questions, you are free to ask them now.

In the event of any problems or concerns/questions later you may contact the researcher at **Email address:** aruna.seerpath@gmail.com, **Telephone number** 0746481598 or the UKZN Humanities & Social Sciences Research Ethics Committee, contact details as follows:

HUMANITIES & SOCIAL SCIENCES RESEARCH ETHICS ADMINISTRATION

Research Office, Westville Campus

Govan Mbeki Building

Private Bag X 54001

Durban 4000 KwaZulu-Natal, SOUTH AFRICA

Tel: 27 31 2604557- Fax: 27 31 2604609

Email: HSSREC@ukzn.ac.za

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

Sincerely

Aruna Seerpath

Thank you for your time

University of KwaZulu-Natal

Researcher: Aruna Seerpath / 0746481598 / aruna.seerpath@gmail.com

Supervisor: Sue Price / 031 260 3162 / pricec@ukzn.ac.za

Project Title: Project managers' and Developers' Perspectives of Software Development Project Success within a South African Bank

Consent to participate

I _____ (full names) have been informed about the study entitled "Project managers' and Developers' Perspectives of Software Development Project Success within a South African Bank" by Aruna Seerpath.

I understand the purpose and procedures of the study.

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

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

If I have any further questions/concerns or queries related to the study, I understand that I may contact the researcher at **Email address:** aruna.seerpath@gmail.com, **Telephone number** 0746481598.

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

HUMANITIES & SOCIAL SCIENCES RESEARCH ETHICS ADMINISTRATION

Research Office, Westville Campus

Govan Mbeki Building

Private Bag X 54001

Durban 4000 KwaZulu-Natal, SOUTH AFRICA

Tel: 27 31 2604557- Fax: 27 31 2604609

Email: HSSREC@ukzn.ac.za

_____ _____

Signature of Participant **Date**

Thank you for your time

Section A Bio-data

Indicate your response by ticking (√) the appropriate checkbox

1.1 What is your gender?

Male	Female

1.2 What is your age?

20-23	24-27	28-31	32-35	36+

1.3 What is your race?

Black	Indian	Coloured	White	Other (please specify):_____

1.4 What is your role?

Project Manager	Software developer

1.5 For how many years have you worked in this role in this bank?

< 4	4 -< 8	8 -<12	12 -<16	16+

1.6 For how many years have you worked in this role in your life?

< 4	4 -< 8	8 -<12	12 -<16	16+

1.7 Did you ever have a different role than the role you are currently in, in this business unit?

Yes	No

1.7.1 **If YES to q1.7**, for how many years were you in that role?

< 4	4 -< 8	8 -<12	12 -<16	16+

1.7.2 **If YES to q1.7**, what was the role? _____

1.8 For how many years have you worked in this business unit?

< 4	4 -< 8	8 -<12	12 -<16	16+

1.9 How many projects do you normally work on at any given time?

0-1	2	3	4	5+

1.10 How many developers are normally in your software development team?

2	3	4	5	6+

1.11 On average how many hours a week do you work?

< 40	40 – 45	46 – 50	51 - 55	56 - 60	61+

1.12 On average how many weekends do you work in a month?

0	1	2	3	4

1.13 In your opinion, how many years of experience should each project manager have for the project to be successful?

<4	4 - <8	8+

1.14 In your opinion, how many years of experience should each software developer have for the project to be successful?

< 4	4 - < 8	8+

Section B

Indicate your agreement that the following items are CRITICAL to the success of a software development project

INDIVIDUAL FACTORS

		Strongly disagree	Disagree	Neutral	Agree	Strongly agree
LSS1	Employees with appropriate skills for the job					
LSS2	Ongoing/available training on skills needed for the job					
LSS3	Problem solving abilities					
LSS4	Knowledge in coding and the software development lifecycle					
EE1	Employees with experience in the software development lifecycle					
EE2	Employees with experience working in a team					
EE3	Employees with 10+ years' experience in coding					
EE4	Employees with 10+ years' in project management					
WLB 1	Ensuring there is a good balance between time at work and personal time					
WLB 2	Working after hours or on weekends					
WLB 3	Feeling happy at work and at home					
JS1	Enjoying the work that I am given					
JS2	Being appreciated for my work					
JS3	Enjoying good relationships with colleagues					
JS4	Having a good working environment					
JS5	Successfully solving problems related to work					

TEAM FACTORS

		Strongly disagree	Disagree	Neutral	Agree	Strongly agree
UR1	A good understanding of the requirements stated in the functional requirement specification					
UR2	A good understanding of the requirements stated in the system requirement specification					
UR3	A good understanding of the screen flows (the design of the screens)					
UR4	Having regular sessions during the course of the project to unpack the requirements of the project					
CR1	Accepting changes requested by users during the project					
CR2	Accepting changes requested by subject matter experts during the project					
CR3	Accepting changes requested by mangers and stakeholders during the project					
QC1	The functionality that is developed is as per the requirement specification					
QC2	The software should work on all platforms i.e. Andriod/ IOS/ Mobi					
QC3	The functionality developed is secure					
QC4	The functionality is easy to use and understand					
QC5	The response times of the app is fast enough					
QC6	The functionality is efficient					
ASM 1	Sufficient subject matter experts (solution/software architects etc.) in the project team					
ASM 2	Having subject matter experts available when necessary					
ASM 3	Not having to wait long for subject matter experts' input					
SP1	Projects that are accurately sized					

		Strongly disagree	Disagree	Neutral	Agree	Strongly agree
SP2	Projects that are accurately scoped					
SP3	Projects with smaller scope and size					
SP4	Projects that are broken into smaller phases					
SP5	Projects that can be completed within a realistic/practical time					
PM1	Meeting all deadlines					
PM2	Staying within the timelines of the system's project plan					
PM3	Delivering the final project on time					
PM4	Staying within budget					
PM5	Incurring minimal additional costs					
PM6	Having an accurate budget					
PM7	Developing and testing all the specified requirements					
PM8	Not leaving out any important functionality					
PM9	Not including any trivial functionality					
RM1	Identifying risks at the beginning of a project					
RM2	Taking precautions to avoid any identified risks					
RM3	Continuously monitoring the project for risks					
RM4	Addressing all risks as soon as they are identified					
MA1	Completing each phase of the software development lifecycle before the next one begins					
MA2	BA's completing their work on time so that the SA's can start theirs					
MA3	SA's completing their work on time before the developers start theirs					

		Strongly disagree	Disagree	Neutral	Agree	Strongly agree
MA4	Developers completing their work on time so that the testers can start theirs					
MA5	Being able to incorporate user input and new requirements at any stage of the development process					
MA6	Having a daily stand-up team meetings to discuss what each member did the day before, what they will do that day, and what problems they face					
MA7	Documenting requirements as they are developed					
MA8	Testing requirements as they are developed					
TD1	Good communication between team members					
TD2	Team members knowing about each other's projects					
TD3	Communicating in person with team members about the projects					
TD4	Regular communication with team members regarding the project					
TD5	All members of the team being treated the same					
TD6	All members of the team being given an equal amount of work					
TD7	Work being allocated in a fair way to all members of the team					
TD8	All members of a team taking responsibility for the project					
TD9	All team members being accountable for the aspect of the project they are working on					
TD10	Managers addressing team members who aren't accountable for their work					
TD11	Receiving financial rewards for doing good work					
TD12	Being motivated as a team to do a good job					

		Strongly disagree	Disagree	Neutral	Agree	Strongly agree
TD13	Receiving encouragement from the team					
TD14	Receiving recognition from the team leader and members of the team for doing good work					
UI1	Ample user involvement during the software development lifecycle					
UI2	Availability of users to work with developers during the project					
UI3	Availability of users to work with project managers during the project					

ORGANISATIONAL FACTORS

		Strongly disagree	Disagree	Neutral	Agree	Strongly agree
SSS1	Managers and stakeholders attending and participating in all necessary project related meetings					
SSS2	Managers and stakeholders providing a solution to problems affecting the project					
SSS3	Management and stakeholders supporting the team					
CBO1	Clearly defined business objectives for each project					
CBO2	Business objectives that provide a sense of purpose and direction					
CBO3	Business objectives that enable the project team to aim towards a common goal					
OC1	A friendly working environment					
OC2	Respecting others' needs for a quiet working environment					
OC3	A supportive working environment					

		Strongly disagree	Disagree	Neutral	Agree	Strongly agree
OC4	Empathy shown by the orgranisation in times of personal need					
OC5	Leadership setting an example for employees by following rules and procedures					
OC6	Following company values					

Section C

1 Indicate your agreement that the following items will ensure software development success

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
1.1 Skilled staff					
1.2 Constantly enhancing skills					
1.3 Project management experience					
1.4 Software development experience					
1.5 Work-life balance					
1.6 Job satisfaction					
1.7 Understanding project requirements					
1.8 Being adaptable to changes to requirements during the project					
1.9 Developing software of a high quality					
1.10 Availability of sufficient subject matter experts					
1.11 Well sized projects					
1.12 Projects being completed on time					
1.13 Projects being within the specified scope					
1.14 Projects remaining within budget					
1.15 Conducting risk assessments					

1.16 Following a structured lifecycle e.g. Waterfall approach					
1.17 Following a flexible lifecycle e.g. Agile approach					
1.18 Good team dynamics (including communication, equality, accountability, motivation)					
1.19 User involvement					
1.20 Support from management and stakeholders					
1.21 Clearly defined business objectives					
1.22 A strong, well defined organisational culture					

2 Indicate your agreement that the following items will lead to software development <u>failure</u>

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
2.1 Lack of skilled staff					
2.2 Lack of training					
2.3 Project managers with no experience					
2.4 Software developers with no experience					
2.5 Not working long hours/on weekends/overtime					
2.6 Being dissatisfied with your job					
2.7 Not understanding requirements					
2.8 Being asked to make changes to requirements at a late stage in the project					
2.9 Producing software of a poor quality					
2.10 Unavailability of subject matter experts					
2.11 Badly sized projects					
2.12 Late completion of projects					
2.13 Projects not meeting requirements					
2.14 Projects going over budget					
2.15 Not conducting risk assessments					
2.16 Following a structured lifecycle e.g. Waterfall approach					
2.17 Not following a flexible lifecycle e.g. Agile approach					
2.18 Lack of team dynamics (including communication, equality, accountability, motivation etc)					
2.19 Lack of user involvement					
2.20 Lack of support from management and stakeholders					
2.21 Unclear business objectives					

2.22 Lack of a strong, well defined organisational culture					

Appendix B: Interview

INTERVIEW SCHEDULE

University of KwaZulu-Natal

Researcher: Aruna Seerpath / 0746481598 / aruna.seerpath@gmail.com

Supervisor: Sue Price / 031 260 3162 / pricec@ukzn.ac.za

Project Title: Project managers' and Developers' Perspectives of Software Development Project Success within a South African Bank

This study has been ethically reviewed and approved by the UKZN Humanities and Social Sciences Research Ethics Committee (approval number_____).

Date: _____

Consent form

Dear colleague,

I am an MCom (IS&IT) student in the School of Management, IT and Governance. Please would you consider participating in my research study which is entitled "Project managers' and Developers' Perspectives of Software Development Project Success within a South African Bank"?

It aims to investigate software development project success from a project managers' and software developers' perspectives within a South African bank The study is expected to include 30 questionnaire participants and 10 interviewees.

Completing the interview will take around 30 minutes.

- ✓ The information you provide will be used to conduct an analysis on software development project success from project managers' and software developers' perspectives.
- ✓ Your name will not be written on the questionnaire or be kept in any other records. We kindly ask you to fill it out with as much accuracy as possible.
- ✓ All responses you provide for this study will remain confidential. When the results of the study are reported, you will not be identified by name or any other information that

could be used to infer your identity. Only researchers will have access to view any data collected during this research, and the responses will not be used for any purposes outside of this study.

- ✓ Your participation is voluntary and you may withdraw from this research any time you wish.
- ✓ Your refusal to participate will not result in any penalty or loss of benefits to which you are otherwise entitled to.
- ✓ All data, both electronic and hard copy will be securely stored during the study and archived for 5 years. After this time, all data will be destroyed.
- ✓ The research intends to abide by all commonly acknowledged ethical codes. You will receive no incentive or payment for your participation.

If you have any questions, you are free to ask them now.

In the event of any problems or concerns/questions later you may contact the researcher at **Email address:** aruna.seerpath@gmail.com, **Telephone number** 0746481598 or the UKZN Humanities & Social Sciences Research Ethics Committee, contact details as follows:

HUMANITIES & SOCIAL SCIENCES RESEARCH ETHICS ADMINISTRATION

Research Office, Westville Campus

Govan Mbeki Building

Private Bag X 54001

Durban 4000 KwaZulu-Natal, SOUTH AFRICA

Tel: 27 31 2604557- Fax: 27 31 2604609

Email: HSSREC@ukzn.ac.za

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

Sincerely

Aruna Seerpath

Thank you for your time

University of KwaZulu-Natal

Researcher: Aruna Seerpath / 0746481598 / aruna.seerpath@gmail.com

Supervisor: Sue Price / 031 260 3162 / pricec@ukzn.ac.za

Project Title: Project managers' and Developers' Perspectives of Software Development Project Success within a South African Bank

Consent to participate

I _____ (full names) have been informed about the study entitled "Project managers' and Developers' Perspectives of Software Development Project Success within a South African Bank" by Aruna Seerpath.

I understand the purpose and procedures of the study.

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

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

If I have any further questions/concerns or queries related to the study, I understand that I may contact the researcher at **Email address:** aruna.seerpath@gmail.com, **Telephone number** 0746481598.

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

HUMANITIES & SOCIAL SCIENCES RESEARCH ETHICS ADMINISTRATION

Research Office, Westville Campus

Govan Mbeki Building

Private Bag X 54001

Durban 4000 KwaZulu-Natal, SOUTH AFRICA

Tel: 27 31 2604557- Fax: 27 31 2604609

Email: HSSREC@ukzn.ac.za

I hereby provide consent to:

Audio-record my interview YES / NO

_____ _____

Signature of Participant **Date**

Interview Questions:

1. What is your current role here?
2. How many years have you worked in this business unit?
3. Have you ever worked in another role within a software development team?
4. Have you ever worked as a developer/project manager at another company or another business unit, if so, how long?
5. Do you feel as though you have all the skills you need to do your job? If not, what kinds of skills would you like to attain?
6. Do you enjoy what you do? If yes, what makes you enjoy it? If no, why?
7. How much overtime do you work and when?
8. Do you think it's company culture to work overtime?
9. Are you rewarded for working overtime? In what ways?
10. Please identify a development project that you think was successful? What was the name of the project?
11. What was the duration of the project?
12. What do you think made it successful?
13. At what stage of the project development life cycle did you consider this project successful?
14. What was the best thing about that project for you, for the team, for the organisation?
15. Please identify a development project that you think was unsuccessful? What was the name of the project?
16. What was the duration of the project?
17. What do you think made it unsuccessful?
18. At what stage of the project development life cycle did you consider this project unsuccessful?
19. What was the worst thing about that project for you, for the team, for the organisation?
20. In your own opinion, what do you think makes a project successful?
21. In your own opinion, what do you think makes a project unsuccessful?
22. What do you think that developers think makes a project:

    a). Successful?

    b). Unsuccessful?

23. What do you think project managers think makes a project:

    a). Successful?

    b). Unsuccessful?

24. Does this business unit have a standard set of success criteria to measure the success of a project? If so, what are some of them?
25. What does this team currently do to prevent project failure?
26. What does organisation currently do to prevent project failure?
27. What do you think needs to change to prevent project failure?
28. How do you think management can support the team more?

29. How do you think management can support the individuals more?
30. What aspects of the project development life cycle are out of your control that negatively affect your job performance?

Appendix C: Interview schedule

Participant	*Date and time of interview*	*Duration of the interview*
Participant 1: Project manager	14 October 2019 10:00 AM	30 minutes
Participant 2: Project manager	14 October 2019 13:00 PM	40 minutes
Participant 3: Software developer	15 October 2019 9:00 AM	20 minutes
Participant 4: Software developer	15 October 2019 10:00 AM	30 minutes
Participant 5: Software developer	16 October 2019 10:00 AM	30 minutes
Participant 6: Software developer	16 October 201 13:00 PM	30 minutes

Appendix D: Research instrument alignment matrix

Section 1

The table below shows the alignment of questions to factors from section A and B of the questionnaire and the questions from the interview schedule.

	Questionnaire questions	**Interview questions**
Individual factors		
Level of skills of staff	LSS1, LSS2, LSS3, LSS4	5
Employee experience	1.5, 1.6, 1.8, 1.13, 1.14, EE1, EE2, EE3, EE4	2, 3, 4
Work-life balance	1.11, 1.12, WLB1, WLB2, WLB3	6, 7, 8
Job satisfaction	JS1, JS2, JS3, JS4, JS5	9, 20
Team factors		
Team dynamics	1.9, 1.10,	
Communication	TD1, TD2, TD3, TD4,	
Equitable / Fair treatment	TD5, TD6, TD7,	
Accountability	TD8, TD9, TD10,	
Motivation	TD11, TD12, TD13, TD14	
Organisational factors		
Support from management and stakeholders	SSS1, SSS2, SSS3	28, 29
Clearly defined business objectives	CBO1, CBO2, CBO3	
Organisational culture	OC1, OC2, OC3, OC4, OC5, OC6	
Project factors		
Understanding requirements	UR1, UR2, UR3, UR4	
Changes to requirements	CR1, CR2, CR3	

Quality of software	QC1, QC2, QC3, QC4, QC4, QC5, QC6
Availability of subject matter experts	ASM1, ASM2, ASM3
Scope and sizing of projects	SP1, SP2, SP3, SP4, SP5
Project management icon triangle	
Time	PM1, PM2, PM3,
Budget	PM4, PM5, PM6,
Scope	PM7, PM8, PM9
Applying risk management	RM1, RM2, RM3, RM4
Methodology adopted	
Traditional	MA1, MA2, MA3, MA4, MA5, MA6, MA7, MA8
Agile	
User involvement	UI1, UI2, UI3

Section 2

The table below shows the ***factors' effects on the software development project success*** from section C of the questionnaire and from the interview schedule.

	Questionnaire	**Interview**
Project success		
Success	1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 1.10, 1.11, 1.12, 1.13, 1.14, 1.15, 1.16, 1.17, 1.18, 1.19, 1.20, 1.21, 1.22	12, 13, 14, 20, 22a, 23a, 24
Project failure		

Failure	2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 2.10, 2.11, 2.12, 2.13, 2.14, 2.15, 2.16, 2.17, 2.18, 2.19, 2.20, 2.21, 2.22	15, 16, 17, 18, 19, 21, 22b, 23b
Project failure prevention		
Project failure prevention		25, 26, 27

Appendix E: References

Ahimbisibwe, A., Daellenbach, U., & Cavana, R. Y. (2017). Empirical comparison of traditional plan-based and agile methodologies. *Journal of Enterprise Information Management*.

Agarwal, N., & Rathod, U. (2006). Defining "success" for software projects: An exploratory revelation. *International Journal of Project Management*, *24*(4), 358–370.

Akgün, A., Keskin, H., Ayar, H., & Okunakol, Z. (2017). Knowledge sharing barriers in software development teams: A multiple case study in Turkey. *Kybernetes*, *46*(4), 603-620. doi: 10.1108/k-04-2016-0081

Aloini, D., Dulmin, R., & Mininno, V. (2007). Risk management in ERP project introduction: Review of the literature. *Information & Management*, *44*(6), 547-567.

Alqahtani, M. A., Alroobaea, R. S., & Mayhew, P. J. (2014). Building a conceptual framework for mobile transaction in Saudi Arabia: A user's perspective. In *2014 Science and Information Conference*. https://doi.org/10.1109/sai.2014.6918303

Arif, B., & Farooqi, Y. A. (2014). Impact of work life balance on job satisfaction and organizational commitment among university teachers: A case study of University of Gujrat, Pakistan. *International journal of multidisciplinary sciences and engineering*, *5*(9), 24-29.

Atkinson, R. (1999). Project management: cost, time and quality, two best guesses and a phenomenon, it's time to accept other success criteria. *International Journal of Project Management*, *17*(6), 337–342.

Attarzadeh, I., & Ow, S. H. (2008). Project management practices: Success versus failure. In *2008 International Symposium on Information Technology*. https://doi.org/ 10.1109/itsim.2008.4631634

Awad, M. A. (2005). A comparison between agile and traditional software development methodologies. *University of Western Australia.* Retrieved from: https://www.unf.edu/~broggio/cen6940/ComparisonAgileTraditional.pdf

Baccarini, D. (1999). The logical framework method for defining project success. *Project Management Journal, 30*(4), 25-32. doi: 10.1177/875697289903000405

Balsamo, S., Di Marco, A., Inverardi, P., & Simeoni, M. (2004). Model-based performance prediction in software development: A survey. *IEEE Transactions on Software Engineering*, *30*(5), 295–310.

Bano, M., & Zowghi, D. (2015). A systematic review on the relationship between user involvement and system success. *Information and Software Technology*, *58*, 148-169.

Bassil, Y. (2012). A simulation model for the waterfall software development life cycle. *International Journal of Engineering & Technology, 2*(5), 7.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., & Kern, J. (2001). Manifesto for agile software development.

Beecham, S. (2014). Motivating software engineers working in virtual teams across the globe. *Software Project Management in a Changing World*, 247-273. doi:10.1007/978-3-642-55035-5_10

Bhattacherjee, A. (2012). *Social Science Research: Principles, Methods, and Practices*. CreateSpace.

Boehm, B., Abts, C., & Chulani, S. (2000). Software development cost estimation approaches. A survey. *Annals of Software Engineering*, *10*(1-4), 177–205.

Boehm, B. W., & Ross, R. (1989). Theory-W software project management principles and examples. *IEEE Transactions on Software Engineering*, *15*(7), 902–916.

Bronte-Stewart, M. (2015). Beyond the iron triangle: Evaluating aspects of success and failure using a project status model. *Computing & Information Systems*, *19*(2), 21-37.

Brooks, F. P. (1975). The mythical man-month. Essays on Software Engineering. Addison-Wesley Publishing Company. 7(1), 93-96. doi:10.1002/net.3230070108

Castillo, J. C., Castro-González, Á., Fernández-Caballero, A., Latorre, J. M., Pastor, J. M., Fernández-Sotos, A., & Salichs, M. A. (2016). Software architecture for smart emotion recognition and regulation of the ageing adult. *Cognitive Computation*, *8*(2), 357-367.

Cerpa, N., & Verner, J. M. (2009). Why did your project fail? *Communications of the ACM*, *52*(12), 130.

Charette, R. N. (2005). Why software fails software failure. *IEEE Spectrum*, *42*(9), 42–49.

Chow, T., & Cao, D. (2008). A survey study of critical success factors in agile software projects. *The Journal of Systems and Software*, *81*(6), 961–971.

Cockburn, A., & Highsmith, J. (2001). Agile software development, the people factor. *Computer*, *34*(11), 131–133.

Crowston, K., Annabi, H., & Howison, J. (2003). Defining open source software project success. *ICIS 2003 Proceedings*, *28*, 3–7.

Daim, T. U., Ha, A., Reutiman, S., Hughes, B., Pathak, U., Bynum, W., & Bhatla, A. (2012). Exploring the communication breakdown in global virtual teams. *International Journal of Project Management*, *30*(2), 199-212.

De Bakker, K., Boonstra, A., & Wortmann, H. (2010). Does risk management contribute to IT project success? A meta-analysis of empirical evidence. *International Journal of Project Management*, *28*(5), 493-503.

Devedzic, V. (2002). Software project mnagement. In *Handbook of Software Engineering and Knowledge Engineering* (pp. 419–446).

Dezdar, S., & Ainin, S. (2011). The influence of organizational factors on successful ERP implementation. *Management Decision*, *49*(6), 911-926.

Dokko, G., Wilk, S. L., & Rothbard, N. P. (2009). Unpacking prior experience: How career history affects job performance. *Organization Science*, *20*(1), 51-68.

Dorsey, P. (2000). Top 10 reasons why systems projects fail. *Retrieved February 10*, 2018. https://www.semanticscholar.org/paper/Top-10-Reasons-Why-Systems-Projects-Fail-Dorsey

Dvir, D., Raz, T., & Shenhar, A. J. (2003). An empirical analysis of the relationship between project planning and project success. *International Journal of Project Management*, *21*(2), 89–95.

Eldrandaly, K. A., Naguib, S. M., & Hassan, M. M. (2015). Evaluation of Critical Success Factors for GIS Implementation Using Analytic Hierarchy Process. *International Journal of Computing*, *4*(3).

Estler, H. C., Nordio, M., Furia, C. A., Meyer, B., & Schneider, J. (2014). Agile vs. structured distributed software development: A case study. *Empirical Software Engineering*, *19*(5), 1197-1224.

Fui-Hoon, N. F., Lee-Shang, L. J., & Kuang, J. (2001). Critical factors for successful

implementation of enterprise systems. *Business Process Management Journal, 7* (3), 285–296.

Gravell, A., Howard, Y., & Aldahmash, A. (2017). A review study on the critical success factors of agile software development.

Grossman, F., Bergin, J., Leip, D., Merritt, S., & Gotel, O. (2004). One XP experience: Introducing agile (XP) software development into a culture that is willing but not ready. In Proceedings of the 2004 Conference of the Centre for Advanced Studies on Collaborative Research. IBM Press, 242–254.

Gu, V. C., Hoffman, J. J., Cao, Q., & Schniederjans, M. J. (2014). The effects of organizational culture and environmental pressures on IT project performance: A moderation perspective. *International Journal of Project Management*, *32*(7), 1170-1181.

Guinan, P. J., Cooprider, J. G., & Faraj, S. (1998). Enabling software development team performance during requirements definition: A behavioral versus technical approach. *Information Systems Research, 9*(2), 101–125.

Halkos, G., & Bousinakis, D. (2010). The effect of stress and satisfaction on productivity. *International Journal of Productivity and Performance Management*, *59*(5), 415-431.

Hastie, S., & Wojewoda, S. (2015). Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch. Retrieved March 13, 2018, from https://www.infoq.com/articles/standish-chaos-2015

Heeks, R., Krishna, S., Nicholson, B., & Sahay, S. (2002). Trajectories and strategies in global software outsourcing relationships. In *Global Perspective of Information Technology Management* 53 - 59.

Hughes, D. L., Dwivedi, Y. K., Rana, N. P., & Simintiras, A. C. (2016). Information systems project failure–analysis of causal links using interpretive structural modelling. *Production Planning & Control*, *27*(16), 1313-1333.

Hughes, D. L., Rana, N. P., & Simintiras, A. C. (2017). The changing landscape of IS project failure: An examination of the key factors. *Journal of Enterprise Information Management*, *30*(1), 142–165.

Hussain, A., & Mkpojiogu, E. O. (2016). Requirements: Towards an understanding on why software projects fail. In *AIP Conference Proceedings* (Vol. 1761, No. 1, p. 020046). AIP Publishing.

Jurison, J. (1999). Software project management: the manager's view. *Communications of the AIS*, *2*(3es), 2–57.

Johri, A. (2011). Sociomaterial bricolage: The creation of location-spanning work practices by global software developers. *Information and Software Technology*, *53*(9), 955-968.

Jørgensen, M. (2016). A survey on the characteristics of projects with success in delivering client benefits. *Information and Software Technology*, *78*, 83-94.

Jureczko, M., & Madeyski, L. (2010). Towards identifying software project clusters with regard to defect prediction. In *Proceedings of the 6th International Conference on Predictive Models in Software Engineering* (p. 9). ACM.

Kapczyński, A. (2016). Critical success factors in managing biometric authentication systems implementation projects. *Zeszyty Naukowe. Organizacja i Zarządzanie/Politechnika Śląska*.

Kappelman, L. A., McKeeman, R., & Zhang, L. (2006). Early warning signs of IT project failure: The dominant dozen. *Information Systems Management*, *23*(4), 31–36.

Kateb, M., Swies, R., Obeidat, B., & Maqableh, M. (2015). An investigation on the critical factors of information system implementation in Jordanian information technology companies. *European Journal of Business and Management*, *7*(36), 11-28.

Kaur, R., & Sengupta, J. (2013). Software process models and analysis on failure of software development projects. *International Journal of Scientific & Engineering Research, 2*(2), 1-4.

Keil, M., Cule, P. E., Lyytinen, K., & Schmidt, R. C. (1998). A framework for identifying software project risks. *Communications of the ACM*, *41*(11), 76–83.

Keil, M., Lee, H. K., & Deng, T. (2013). Understanding the most critical skills for managing IT projects: A Delphi study of IT project managers. *Information & management*, *50*(7), 398-414.

Kivelä, A. (2019). Investigating gender equality and gender-based discrimination in software development.

Khoza, L., & Marnewick, C. (2020). Waterfall and Agile information system project success rates-A South African perspective. *South African Computer Journal*, *32*(1), 43-73.

Korrapati, R., & Eedara, V. S. (2010). A study of the relationship between software project success and employee job satisfaction. In *Allied Academies International Conference. Academy of Management Information and Decision Sciences. Proceedings* (Vol. 14, No. 1, p. 22 - 25). Jordan Whitney Enterprises, Inc.

Kramer, M. (2018). Best Practices in Systems Development Lifecycle: An Analysis Based on the Waterfall Model. *Review of Business & Finance Studies*, *9*(1), 77-84.

Larsson, S., & Ekdahl, F. (2005). In Proceedings of the ESEIW 2003 workshop on empirical studies in software engineering WSESE 2003, Rome, Italy, September 2003. *Improving Software Product Integration*, 33.

Layman, L., Williams, L., Damian, D., & Bures, H. (2006). Essential communication practices for Extreme Programming in a global software development team. *Information and Software Technology*, *48*(9), 781–794.

Lech, P. (2013). Time, budget, and functionality? IT project success criteria revised. *Information Systems Management*, *30*(3), 263-275.

Lee, J. C., Shiue, Y. C., & Chen, C. Y. (2016). Examining the impacts of organizational culture and top management support of knowledge sharing on the success of software process improvement. *Computers in Human Behavior*, *54*, 462-474.

Lehtinen, T. O., Mäntylä, M. V., Vanhanen, J., Itkonen, J., & Lassenius, C. (2014). Perceived causes of software project failures–An analysis of their relationships. *Information and Software Technology*, *56*(6), 623-643.

Linberg, K. R. (1999). Software developer perceptions about software project failure: A case study. *The Journal of Systems and Software*, *49*(2-3), 177–192. 44

Lyu, M. R. (2007). Software reliability engineering: A roadmap. *Future of Software Engineering (FOSE 07)*,153-170. doi:10.1109/fose.2007.24

Mäntylä, M. V., Jørgensen, M., Ralph, P., & Erdogmus, H. (2017). Guest editorial for special section on success and failure in software engineering. *Empirical Software Engineering*, *22*(5), 2281–2297.

Marques, R., Costa, G., Silva, M., & Gonçalve, P. (2017). A survey of failures in the software development process. *ECIS 2017 Proceedings*, 2445-2459.

McHugh, O., Conboy, K., & Lang, M. (2011). Agile practices: The impact on trust in software project teams. *IEEE Software*, *29*(3), 71-76.

McIntosh, S., Kamei, Y., Adams, B., & Hassan, A. E. (2016). An empirical study of the impact of modern code review practices on software quality. *Empirical Software Engineering*, *21*(5), 2146-2189.

Mishra, D., & Mishra, A. (2011). A review of non-technical issues in global software development. *International Journal of Computer Applications in Technology*, *40*(3), 216-224.

Misirli, A. T., Verner, J., Markkula, J., & Oivo, M. (2014). A survey on project factors that motivate Finnish software engineers. In *2014 IEEE Eighth International Conference on Research Challenges in Information Science (RCIS)*. https://doi.org/ 10.1109/rcis.2014.6861052

Mtsweni, E. S., Hörne, T., & van der Poll, J. A. (2016). Soft skills for software project team members. *International Journal of Computer Theory and Engineering*, *8*(2), 150.

Munassar, N. M. A., & Govardhan, A. (2010). A comparison between five models of software engineering. *International Journal of Computer Science Issues*, *7*(5), 95–101.

Munns, A. K., & Bjeirmi, B. F. (1996). The role of project management in achieving project success. *International Journal of Project Management*, *14*(2), 81–87.

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

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, *48*(5), 72–78.

Neves, F. G., Borgman, H., & Heier, H. (2016). Success lies in the eye of the beholder: The mismatch between perceived and real IT project management performance. In *2016 49th Hawaii International Conference on System Sciences (HICSS)* (pp. 5878-5887). IEEE.

Nofal, M. I. M., & Yusof, Z. M. (2016). Conceptual model of enterprise resource planning and business intelligence systems usage. *International Journal of Business Information Systems*, *21*(2), 178-194.

Norvig, P. (2001). Teach yourself programming in ten years. Retrieved 10 January 2020. URL http://norvig.com/21-days.

Octavianus, R., & Mursanto, P. (2018). The analysis of critical success factor ranking for software development and implementation project using AHP. In *2018 International Conference on Advanced Computer Science and Information Systems (ICACSIS)* (pp. 313-318). IEEE.

Pathak, R. R. (2018). Work Life Balance in Nepalese Commercial Banks. *Journal of Business and Social Sciences*, *2*(1), 116-125.

Peat, M., & Franklin, S. (2002). Use of online and offline formative and summative assessment opportunities: have they had any impact on student learning?. In *ASCILITE* (pp. 505-513).

Paetsch, F., Eberlein, A., & Maurer, F. (2003). Requirements engineering and agile software development. *Enabling Technologies: Infrastructure for Collaborative Enterprises* (pp. 308–313). Twelfth IEEE International Workshops.

Perera, I. (2011). Impact of poor requirement engineering in software outsourcing: A study on software developers' experience. *International Journal of Computers Communications & Control*, *6*(2), 337-348.

Phellas, C. N., Bloch, A., & Seale, C. (2011). Structured methods: Interviews, questionnaires and observation. *Researching Society and Culture*, *3*.

Poon, P., & Wagner, C. (2001). Critical success factors revisited: success and failure cases of information systems for senior executives. *Decision Support Systems*, *30*(4), 393-418.

Procaccino, D. J., Verner, J. M., Overmyer, S. P., & Darter, M. E. (2002). Case study: Factors for early prediction of software development success. *Information and Software Technology*, *44*(1), 53–62.

Püschel, J., Mazzon, J. A., & Hernandez, J. M. C. (2010). Mobile banking: Proposition of an integrated adoption intention framework. *International Journal of Bank Marketing*, *28*(5), 389–409.

Rehman, R. (2006). Factors that contribute to open source software project success. 1-205. doi:10.22215/etd/2006-08297

Reed, A. H., & Knight, L. V. (2010). Project risk differences between virtual and co-located teams. *Journal of Computer Information Systems*, *51*(1), 19-30.

Robinson, H., & Sharp, H. (2005). Organisational culture and XP: Three case studies. In Agile Conference Proceedings IEEE, 49–58.

Rezvani, A., Chang, A., Wiewiora, A., Ashkanasy, N. M., Jordan, P. J., & Zolin, R. (2016). Manager emotional intelligence and project success: The mediating role of job satisfaction and trust. *International Journal of Project Management*, *34*(7), 1112-1122.

Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, *14*(2), 131–164.

Ruparelia, N. B. (2010). Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes*, *35*(3), 8–13.

Savolainen, P., Ahonen, J. J., & Richardson, I. (2012). Software development project success and failure from the supplier's perspective: A systematic literature review. *International Journal of Project Management*, *30*(4), 458-469.

Sekaran, U., & Bougie, R. (2016). Research methods for business: A skill building approach. *John Wiley & Sons.*

Serrador, P., & Pinto, J. K. (2015). Does Agile work? —A quantitative analysis of agile project success. *International Journal of Project Management*, *33*(5), 1040–1051.

Sharma, S., Sarkar, D., & Gupta, D. (2012). Agile processes and methodologies: A conceptual study. *International Journal on Computer Science and Engineering*, *4*(5), 892.

Shenhar, A. J., & Dvir, D. (2007). Reinventing project management: The diamond approach to successful growth and innovation. Boston, MA: Harvard Business School Press.

Sheffield, J., & Lemétayer, J. (2013). Factors associated with the software development agility of successful projects. *International Journal of Project Management*, *31*(3), 459-472.

Shahibi, M. S., Sarifudin, S. A., Hussin, N., Ibrahim, Z., Ali, J., & Fakeh, S. K. W. (2019). Factors influencing information technology project management success in the financial industry. *Journal of Theoretical and Applied Information Technology, 97*(10).

Stellman, A., & Greene, J. (2005). Applied software project management. Sebastopol, CA: O'Reilly Media, Inc.

Stoica, M., Mircea, M., & Ghilic-Micu, B. (2013). Software development: Agile vs. traditional. *Informatica Economica*, *17*(4), 64.

Sukhoo, A., Barnard, A., Eloff, M. M., Van der Poll, J. A., & Motah, M. (2005). Accommodating soft skills in software project management. In *Proceedings of the 2005 InSITE Conference*. https://doi.org/10.28945/2923

Suma, V., & Nair, T. R. (2010). Effective defect prevention approach in software process for achieving better quality levels. *arXiv preprint arXiv:1001.3552*.

Takpuie, D., & Tanner, M. (2016). Investigating the characteristics needed by scrum team members to successfully transfer tacit knowledge during agile software projects. *Electronic Journal of Information Systems Evaluation*, *19*(1), 36-54.

Tarawneh, H. (2011). A suggested theoretical framework for software project success. *Journal of Software Engineering and Applications*, *04*(11), 646–651.

Thakurta, R. (2013). Impact of Scope Creep on Software Project Quality. *Vilakshan: The XIMB Journal of Management*, *10*(1).

Theocharis, G., Kuhrmann, M., Münch, J., & Diebold, P. (2015, December). Is water-scrum-fall reality? On the use of agile and traditional development practices.

In *International Conference on Product-Focused Software Process Improvement* (pp. 149-166). Springer, Cham.

Tornjanski, V., Marinkovic, S., Levi Jaksic, M., & Bogojevic Arsic, V. (2015). The prioritization of open innovation determinants in banking. *Industrija*, *43*(3).

Uskov, V. (2017) Software Requirements Engineering Education in a Changing world.

Verner, J. M., & Cerpa, N. (2005). Australian software development: what software project management practices lead to success? In *2005 Australian Software Engineering Conference* (pp. 70-77). IEEE.

Verner, J. M., & Evanco, W. M. (2003). An investigation into software development process knowledge. In *Managing Software Engineering Knowledge* (pp. 29–47).

Verner, J. M., & Evanco, W. M. (2005). In-house software development: What project management practices lead to success? *IEEE Software*, *22*(1), 86–93.

Verner, J., Sampson, J., & Cerpa, N. (2008). What factors lead to software project failure? In *2008 Second International Conference on Research Challenges in Information Science*. https://doi.org/10.1109/rcis.2008.4632095

Wallace, L., Keil, M., & Rai, A. (2004). How software project risk affects project performance: An investigation of the dimensions of risk and an exploratory model. *Decision Sciences*, *35*(2), 289–321.

Wateridge, J. (1998). How can IS/IT projects be measured for success? *International Journal of Project Management*, *16*(1), 59–63.

Wasserman, T. (2010). Software engineering issues for mobile application development.

Whittaker, B. (1999). What went wrong? Unsuccessful information technology projects. *Information Management & Computer Security*, *7*(1), 23–30.

Xia, X., Lo, D., Bao, L., Sharma, A., & Li, S. (2017). Personality and project success: Insights from a large-scale study with professionals. In *2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (pp. 318-328). IEEE.

Yeoh, W., & Koronios, A. (2010). Critical success factors for business intelligence systems. *Journal of Computer Information Systems*, *50*(3), 23-32.

Zahra, K., Azam, F., Ilyas, F., Faisal, H., Ambreen, N., & Gondal, N. (2017). Success factors of organizational change in software process improvement: A systematic literature review. In *Proceedings of the 5th International Conference on Information and Education Technology* (pp. 155-160).

Appendix F: Ethical clearance

UNIVERSITY OF KWAZULU-NATAL
INYUVESI YAKWAZULU-NATALI

21 April 2021

Miss Aruna Seerpath (213503484)
School of Management, IT & Governance
Westville Campus

Dear Miss Seerpath,

Protocol reference number: HSSREC/00000002/2019
Project title: Project managers and developers perspectives of software development project success within a South African bank
Amended title: Software development project success: Perspectives of project managers and developers in a South African Bank

Approval Notification – Amendment Application

This letter serves to notify you that your application and request for an amendment received on 09 April 2021 has now been approved as follows:

- Change in title

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

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

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

Best wishes for the successful completion of your research protocol.

[redacted]

.....
Professor Dipane Hlalele (Chair)

/ms

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

Appendix G: Editors Letter

25 Maple Crescent
Circle Park
KLOOF
3610

Phone 031 – 7075912
0823757722
Fax 031 - 7110458
E-mail:
dr1govender@telkomsa.net
sathsgovender4@gmail.com

Dr Saths Govender

14 NOVEMBER 2020

TO WHOM IT MAY CONCERN

LANGUAGE CLEARANCE CERTIFICATE

This serves to inform that I have read the final version of the dissertation titled:

Project managers' and developers' perspectives of software development project success within a South African bank

by Aruna Seerpath, student no.213503484.

To the best of my knowledge, all the proposed amendments have been effected and the work is free of spelling and grammatical errors. I am of the view that the quality of language used meets generally accepted academic standards.

Yours faithfully

[redacted]

DR S. GOVENDER

B Paed. (Arts), B.A. (Hons), B Ed.

Cambridge Certificate for English Medium Teachers

MPA, D Admin.