The influence of software development project maturity levels on software project outcome

by

MOKETO EPHRAIM BOGOPA

A dissertation submitted in fulfilment for the Degree

of

Master's in Commerce

in

Information Technology Management

at the

College of Business and Economics

UNIVERSITY OF JOHANNESBURG

Supervisor: Prof C Marnewick

2019

DECLARATION

Student number: 201477245

I certify that the *dissertation* submitted by me for the degree *Master's of Commerce (Information Technology Management)* at the University of Johannesburg is my independent work and has not been submitted by me for a degree at another university.

_MR MOKETO EPHRAIM BOGOPA________________________________
(Name in block letters – no signature)

DEDICATION

This dissertation represents a significant learning journey in my life.

I dedicate this work to my lovely son **Previn Mochabo Bogopa** and my lovely daughter, **Aurelia Maloko Mpho Bogopa**, who was born during my complex schedule.

I am indebted to my darling wife, **Emina Bogopa**, and our kids for tolerating my absence from home during my studies. **Emina** thank you for your enduring support and encouragement, and for always looking after the kids during my absence.

I would also like to thank the following people, in no specific order of importance:

- To my father, **Ntate Simon Bogopa**, from whom I drew inspiration, your constant prayers are working miracles in my life. You are the best.
- To my mother, **Mme Merriam Bogopa**, for your care and love. Thanks for raising me to be the man I am today.
- To my parents, who despite not being educated, are very supportive on issues related to education. My parents would rather go hungry and take someone to school.
- To my sister **Agnes Bogopa**, thank you for the foundational support in my education.
- A big thank you to my mother-in-law, **Mme Kgolane Mapadimeng**, for playing a role in taking care of my child (*Aurelia Maloko*) when I was busy with my studies.
- Thank you to my father-in-law, **Ntate Peter Mapadimeng**, for borrowing on my behalf books from libraries of your institution, Tuks (University of Pretoria), when some of the books were not available at the Universities of Johannesburg and South Africa.

ACKNOWLEDGEMENTS

I wish to thank the following people:

To my Supervisor, thank you for giving me direction to continue my dissertation. Professor Carl Marnewick I thank you for your patience with me throughout my research journey. I am grateful to be supervised by such a great supervisor. If I had to write a book about my supervisor, it will be 1000 pages long. University of Johannesburg, I express my sincere gratitude for the knowledge I have gained.

Members of my faculty at my institution (UNISA School of Computing):

Prof Ernest, thank you for your indispensable input that improved the paper significantly and for your time that you regularly take from your busy schedule to accommodate me. I have a great rapport with my local mentor: Prof Ernest, I appreciate his guidance on this dissertation.

Mr Samuel Mtsweni and Ms Portia Buthelezi for playing a major role when designing my research instrument. A special thanks to Dr Mathias Mujinga for helping a lot with of Microsoft word formatting and the layout configurations of the document. Not forgetting Prof. P. Mkhize, thank you so much for your efforts and helping me to realize the main objectives of my dissertation. Also for inviting me to present my literature review to our research group of software engineering so that they can come with inputs and critical analysis if possible. How can I forget Dr Blessings Tawanda Chiyangwa, who was very helpful with statistics analysis guidance. By the way, I wish to thank them the same.

ABSTRACT

The study investigates the influence of maturity level on the outcomes of Software development projects. Other objectives of the research conducted covers the factors that makes a software project to succeed or fail. The study also discusses the maturity levels of each development phase of each software development team, organization, or company that develop software code. The research determines other things like the correlation between factors and software development projects. The aim of the study is to determine if the maturity level plays a major role on the outcome of the software development project. The results can be used with confidence because the reliability and validity tests were proven. Studies such as the Standish Group which was published in 2013 and 2014 and the Prosperus report (2003, 2008, and 2013), highlight the frequency with which Software development project failure occurs and the link between project success or lack thereof and project management maturity. However, this research has revealed that software projects are currently succeeding as viewed by members of software development teams. The success metrics were initially defined as meeting time, budget and scope. The definition is evolving around the three main traditional measures of project success. The research has revealed that the projects that are on time, work well, have excellent quality, and have happy customers. Our statistical analysis indicated that Software development projects are doing well in South Africa; European countries can insource their project to South Africa. The study was quantitative and implemented the survey in the form of structured questionnaire. As nature of cross-sectional study, the data was collected once from members of software development teams. A survey conducted with 111 software developers; 18 Business analysts; 13 project managers and other 67 members of software development team. This research has employed CMMI maturity model in order to determine the maturity level of the software development process, project management processes and organisational processes of the IT organisation. The research has found a medium strength of relationship of correlation between project success and maturity levels, and another correlation between project success/outcome and critical success factors. The research findings confirmed that maturity level influences the success rate of software development project. Software project maturity performance was measured by five constructs: requirement management, requirement development, technical solution, product integration, and verification. While project management maturity performance was measured by six constructs: project planning, project monitoring and control, supplier agreement management, risk management and quantitative project management. Similarly organisational perceived performance was measured by 5 constructs: organisational process focus, organisation process definition, organisational training, organizational process performance and organization performance management.

Key Terms

Organisational Maturity; Software development project maturity; Project management maturity; Project Management Maturity Model; Project outcomes; Critical factors; Software development project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ACRONYMS

AIPM	Australian Institute of Project Management
APMBoK	Association of Project Management Body of Knowledge
BCS	British Computer Society
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
CMMI-DEV	Capability Maturity Model Integration for Development
DBA	Database Administrator
IEEE	The Institute of Electrical and Electronics Engineers
IS	Information System
IT	Information Technology
ITPM	Information Technology Project Management
IT Governance	Information Technology Governance
KPI	Key Performance Indicators
OGC	Office of Government Commerce
MS	Microsoft
PI	Product Integration
PM	Project Manager/Management
PMBoK®	Project Management Body of Knowledge
PMMM	Project Management Maturity
PMI	Project Management Institute
PMIS	Project Management Information System
PMO	Project Management Office
PMSA	Project Management South Africa
PRINCE2	Project IN Controlled Environment Version 2
P2MM	PRINCE2 Maturity Model

P-CMM	People Capability Maturity Model
RD	Requirements Development
ROI	Return on investment
RUP	Rational Unified Process
SA	South Africa
SE-CMM	Systems Engineering Capability Maturity Model
SEI	Software Engineering Institute
SMART	Specific, Measurable, Attainable, Relevant, and Time-bound
SPICE	Software Process Improvement and Capability dEtermination
SPSS	Statistical Package for the Social Sciences
SW-CMM	Capability Maturity Model for Software or Software Capacity Maturity Model
TS	Technical Solution
TSP	Team Software Process
UML	Unified Modelling Language

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND

A project is described as a temporary endeavour undertaken to accomplish a unique purpose (Marchewka, 2016). The reason why a project is regarded as temporary is because it has start and end periods. To finish the project, it requires determination from project members. Therefore, its scope is limited and defined by specific stakeholders.

Information system projects share many similarities with generic projects because they consist of activities, with each activity having a duration and requiring resources (Olson, 2014). Some of information system projects make them distinctly different from generic projects (Dorsey, 2005; Olson, 2014). Information Systems project types are served by a standard methodology with the need to identify user requirements that are followed by construction of a system, producing and delivery of the working system, training and implementation, and, ultimately, maintenance of the system (Olson, 2004:5). When a building is half-done, there is a visible progress, but when a software project is half-done, there is very little to see (Dorsey, 2005). Information system projects involve a development environment called software development.

The generic definition refers to software as a computer program that is used by a computer user to perform different tasks. Ruhe & Wohlin (2016) defines software as a product of the cognitive processes of individuals who engage in innovative collaboration. The definition of the word "*software*" depends on the field of specialisation; all practitioners define it to suit their environment. In other fields, software is regarded as the source code of programming language.

New software continues to make significant contributions to society at large. For example, one of the most recent advanced software that is currently revolutionising the automobile industry is the Tesla "autopilot" software. This advanced driver assist software system gives vehicles semi-autonomous navigation capabilities that allows them to, for example, change lanes, "see" other cars around them, self-park, and enter and exit highways. Software development, which is the focus of this research study, is the fundamental activity where the software is designed and programmed (Sommerville, 2011).

All the Information Technology (IT) projects related terms should be defined so that the difference between information system, software development and Information and Communication Technology (ICT) projects can be realised. Sommerville (2011) defines an information system as a system with a primary purpose of managing and providing access to a collection of information. Examples of an information system include access to a library catalogue, a flight time-table, or the records of patients in a hospital. A software project is on the other hand defined by Ahmed (2012) as a software development, software customization, software integration, software maintenance, or just one phase of the software development life-cycle. Software project life-cycles are models of how software projects pass through the phases of development, from their initiation to their closure (Ruhe and Wohlin, 2016), and they are broadly consisted of the following activities: requirements analysis, software coding and testing a software end product.

Software projects and the management thereof are slightly different from information system projects and other projects in a number of ways. The focus of this research study is on software development, not the entire information system.

The current software marketplace is occupied by small software companies (Larrucea *et al.*, 2016), but many people still think about large organisations only when they think about technology innovation. A software development project, which is called a software project for the sake of simplicity, has aim of developing a software product or maintaining a developed software (Chemuturi & Cagley, 2010). Software development projects require skills and expertise to use programming languages. Information Technology is a combination of software system and the computer hardware in which the software will run (Ahmed, 2012). Chemuturi & Cagley (2010) differentiate software projects from other types of projects, below are the definitions used to clarify the difference:

- Software project has a starting and end date.
- Delivered product is a functional software that can be tested, and related artefacts.
- Software project has activities in each phase of the software life circle, including during handover.
- Activities that are excluded are those activities not performed by software development team like project acquisition.

At the recent Forbes Reinventry American Summit, Ford executive chairman and great-grandson of Henry Ford, Bill Ford, mentioned that semi-autonomous cars are imminent. Bill Ford also believes the technology will arrive sooner than later before society actually figures out how to make it work (www.ford.com). Such projects fall under the IT projects. According to Schwalbe (2014), IT projects a very diverse and software development projects is part of IT diverse projects. The IT projects are development, implementation and infrastructure (Kabir and Rusu, 2016), and software development projects might include building a simple, standalone Microsoft Access or Excel application, global electronic commerce that uses C++ or .Net and runs on many platforms like android or apple (Schwalbe, 2014). Although an autonomous car project has proven to be a challenge, society, potential customers and the IT and car manufacturing industry are eagerly waiting for the finalisation of this project. Due to technological advancements in software development, the project will be delivered timeously, and traffic departments should be prepare and budget to acquire robots to issue traffic fines to self-driving cars upon violation of traffic laws.

1.2 PROBLEM STATEMENT

The most important challenge faced by software development researchers and professionals is the low success rate of software development projects. While small and medium software companies generally aim for big annual profits amounting to millions, the challenge that is discouraging from investing in software development projects is the massive rate of failed software projects that are regularly reported on IT management magazines.

Successes, challenges and failure rates on information system and software development projects have been a subject of discussion for several years (Ahmed, 2012; Dorsey, 2005). The rate of software project failure is high when compared with other advanced technological projects (Yeo, 2002). The software development projects have 'failed', in the areas of budget and/or schedule overruns and/or for not meeting users' requirements (Yeo, 2002:241). Yeo (2002) has also predicted that IS projects would continue to be 'challenged' or 'impaired'. The Information Technology/software industry in the world is delivering higher technology enhanced products and services everyday across many industries. Many project management or software maturity models are dominating different industries and countries. A software development project outcome can be classified as a challenge, failure or success.

In his latest book, Marchewka (2013) has professed that "although IT is becoming more reliable, faster, less expensive, but the costs, complexity, and risks of managing IT projects continues to be a challenge for many organisations". Marchewka (2013) has also raised concerns about projects that do not receive any funding; such projects will either have to wait or fall by the wayside. The decision to sponsor or finance an IT/IS or software project should be based on the return of profit and benefits that the completed product will deliver back to the organization. One of the most important issues for organizations and Information technology professionals is the success rate of software projects. Therefore, the aim of this research study is to determine whether maturity level contributes to the success of the software development project.

Surveys results are regularly published by IT Cortex and provide the statistical information regarding the rate of failure in software development projects (Qassim, 2008:12). The following are lists of available surveys that present different figures about software projects results (Qassim, 2008:12;Majeed *et al.*, 2013):

1. The Robins-Gioia survey (2001)
2. The Conference Board survey (2001)
3. The KPMG Canada survey (1997)
4. The CHAOS Report (1995)
5. The OASIG survey (1995)
6. PricewaterhouseCoopers (2004)
7. Pulse of the profession (PMI) (2006)

The surveys produce the reports of the above-mentioned longitudinal studies at different periods. While a nominal fee is charged for some of these reports, other providers make them available to the public at no cost. All the relevant data and findings for software development projects must be accessible to the general public and the community at large. Although few studies by, to name a few, Carl Marnewick, Mariki Eloff, Erasmus and Les Labuschagne have contributed to research in software projects, research in IT software projects within the South African context is generally lacking (Joseph et al., 2016). The Prosperus Report has since 2003 been monitoring success rates of IT project in South Africa using iron triangle as a definition of success.

1.3 DISSERTATION RESEARCH AIM

The main aim of this research study is to determine whether project success rate is influenced by the level of software project management maturity.

1.4 DISSERTATION RESEARCH QUESTIONS

The following 2 dissertation research questions are proposed:

1. What is the software development project success rate?

2. What is the impact of software development maturity level on project outcome?

1.5 DISSERTATION RESEARCH OBJECTIVES

In order to address the proposed research questions, the following 3 objectives were set:

1. To determine the software development project success rate, and the factors that affect the success of those projects.
2. To investigate the maturity level of software development organisations.
3. To analyse the impact of IT/software development project management maturity models and levels on software project outcome.

1.6 DISSERTATION SCOPE (DELINEATIONS AND LIMITATION)

Although the scope of the dissertation is limited to software development in Information Technology/software sector, it is also applicable to IT projects in any environment including financial, construction, chemical, mining, retail, and other engineering industries. Since the area of IT and ICT is very broad, the focus of the study is specifically on software development. In this research study, distinction is made between ICT, IT and software development projects, although ICT and IT involves software partially.

1.7 DEFINITION OF TERMS

The following terms and concepts will be used throughout the dissertation, especially in Chapter 2 (i.e. literature review). Some of the terms will be explained when they appear in the body of the study, some are summarised in **Table 1.1**.

Table 1. 1: *Table of definitions*

Term	Definition
Factors	Is a contributor
Project	is a temporary endeavour undertaken to accomplish a unique purpose (Marchewka, 2016).
Process areas	is a collection of one or more specific goals within an organisation (Persse, 2007).
Project team	A small number of people who are committed to a common purpose, goals and directly accountable to the project assigned to.
Development team	A group of individuals who specialises in different skills of the same industry or environment, in software industry we refer to software

	testers, software developers, software architects, project managers, system analysts, business analysts and etc.
Success factors	As defined by Association for Project Management (2012:32): are management practices that, when implemented, will increase the probability of success of a project.
Maturity	Is the quality or state of being mature.
Capability	Ability of the organisation to produce the products predictably and consistently.

1.8 ASSUMPTIONS

According to Hofstee (2006), all the research designs contain assumptions as the essential part of the research. Assumptions are fundamental to research. The researcher believes that the questionnaire will be completed by the members of software development team. The team members are members that take part in software development.

1.9 SIGNIFICANCE OF THE RESEARCH

This research examines the benefits of maturity levels of software development organisations/teams, so that the customer can expect a quality software product from the supplier based on the current maturity status. The study uses quantitative methodological approach so that empirical evidence can be rich. The research will contribute largely on software development organisations.

1.10 DISSERTATION LAYOUT

This dissertation is divided into five chapters, which are structured as follows:

- **Chapter One** is an introductory part of the research. After outlining background information relating to the study, the chapter introduces relevant literature and an overview of the dissertation. The chapter has differentiated software project from other IT related, ICT and Information system projects, and maintained the focus of the study.
- **Chapter Two**, which presents the published literature on software development, project management front publishers that have written the papers that have changed the world of Information Technology/Softwares. The international leaders of IT Project management from abroad industry to local, considering Harold Kerzner, Les

Labuschagne, Carl Marnewick, J. Cent Crawford, Watts Humphrey and Ernest Mnkandla.

The literature review classified into different sections. After discussing the project success rate in the first section, the second section unpacks project management maturity. The last section is focused on maturity level of software development projects. The chapter represent and discloses what is already known about the project success and software maturity models.

- **Chapter Three**: In chapter three, the research design and methodological approaches adopted for investigating whether the project outcome is determined by maturity level are discussed in to details. In a nutshell, Chapter 3 describes how the study about maturity and project success rate was conducted and how data will be collected, and how results will be analysed to determine and support the claim that the maturity level influence the project success rate. The research methods that are adopted to find out the factors that affect software projects are also discussed.
- **Chapter Four**: This chapter presents the study results and discusses the relevant statistical tests that will be used to determine the outcome of the projects and the role played by projects success factors. The chapter prescribes what must be done with the collected data. The information collected is analysed to determine the level of perceived maturity level against the software development project outcome.
- **Chapter Five:** Chapter 5 discusses the future research directions about the software maturity models, software project success rate, and the factors that influences the project outcome, and also summarizes the overall results of the study. The chapter also reveals to the reader what was discovered about software development project outcomes and suggests the areas that require further attention.

1.11 CITATION MANAGEMENT AND REFERENCE METHOD

References were managed electronically with the Mendeley citation manager. Mendeley offers approximately 6000 citation styles. For consistency the Harvard method of referencing, British standard BS ISO 690:2012, was used throughout this dissertation.

The next chapter is a collection of literature material for the subject that was introduced in chapter 1. The chapter also serves to test whether the researcher has a comprehensive and good knowledge of all the scholarly work relating to the subject matter.

CHAPTER 2: LITERATURE REVIEW

2.1 INTRODUCTION

Donaldson & Siegel (1997) define a software project as a planned undertaking whose purpose is to produce a system or systems with software content. Every software development team and software providers would like to deliver software projects that satisfy a customer and relevant stakeholders. It is every software development team's desire to deliver a successful software project. In this study, a software development team refers to the team members that develops software, a team member could be coding the software with one or a combination of programming languages, other team members can design technology architecture, analyse the software requirements, test the completed software product or implement the final product. The size and responsibility of software development team varies according to the size of the organisation and the scope of the software under development. In this study, the term "organisation" is used to refer to company, division or department, or corporation. A team of more than one individual undertaking software projects forms part of the scope of this study.

Most projects, including construction projects, contain an IT component or a piece of software application, the project may require the implementation of IT elements on some stage. The technology application, which applies across the projects, is the backbone of any project. This chapter is focussed on reviewing the literature relating to critical success and failure factors that contribute towards or drive software development projects as well as the impact of software development maturity level on software development project teams. Specifically, the study examines the current drivers of software development project successes, and the impact of software development maturity level to software development projects. The literature review outlines all the contributors to failure and success, also the effect of maturity level on software development projects teams.

In order to achieve the main aim of the literature review, the following chapter objectives need to be discovered:

1 Determine current software project success rates; and
2 Determine the relationship between success and maturity, and
3 Determine IT project management maturity level, specifically software development.

This literature review is divided into three sections, namely: project success measurements, project management maturity and maturity levels of software development projects. The software development project as a whole is measured as if it was delivered on time, within budget, meets quality specified and the agreed scope. A strategic approach towards the undertaking of this literature review is illustrated in **Figure 2.1**. The three above-mentioned components of the literature review are discussed individually and in more detail in the sections that follows.

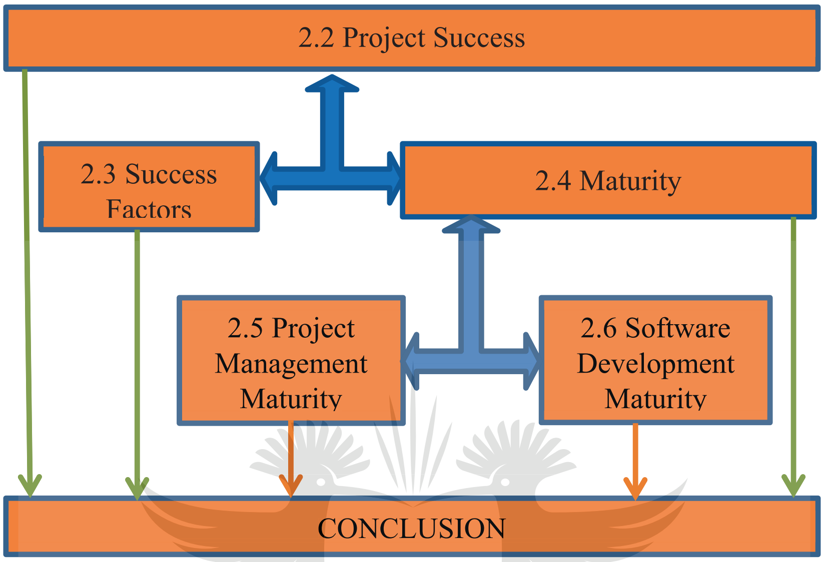


Figure 2. 1: *A pictorial illustration of the literature review chapter*

2.2 PROJECT SUCCESS

Successful software projects are very few in the software industry (Poranen, 2014). One of the possible reason is that the definition of "success" for software projects is different for different people and stakeholders (Marnewick, 2013a:20; Poranen 2014). Software development projects have three possible outcomes, the project is successful, a failure or considered a challenged project. Defining a successful project is a challenge within software industry. According to Pretorius, Steyn & Jordaan (2012) the definition of the "*project success*" depends on a person's perspective. Similar sentiments were echoed by Burke (2011) when suggesting that the project success depends on whose perspective is considered between the manager and project sponsor, because both have different success metrics for defining project success. Traditionally, projects are declared successful when they meet three important goals, namely: scope, time, and cost. Project success is measured by comparing the outcome of the above goals. Failure to meet one of the goals lead to project failure or the project is categorised as challenged. The word "*scope*" refers to a combination of quality and functionality (Poranen, 2014). Collins Dictionary defines success as: the favourable outcome of something attempted;

or the attainment of wealth, fame, etc. According to Camilleri (2016), the word success is very illusive when applied to projects, because even the current successful project can be declared a failure in future.

Projects that are completed on time, within budget and according to the scope defined by stakeholders of the organisation are regarded as being successful (Kaur and Aggrawal, 2013:76). According to Humble & Russell (2009), the definition of the phrase '*successful projects*' is not enough if it does not include the concepts of completing the project on time and in budget. Also, software development researchers define the success of a project differently. "A *project that is perceived as success by a project manager and team members might be perceived as a failure by the client*" (Belassi and Tukel, 1996:141). The definition of the phrase '*project success'* has changed in the last 20 years and this is expected to change even further in the next 10 years. Researchers and practitioners simply have to come to terms with the fact that there will never be uniformity with regards to the definition of "*project success*".

Prabhakar (2008) has a simple definition of project success that hinges on three success metrics, namely: completed on time, within budget, and meets performance requirements. Agarwal & Rathod (2006) defines the software project success as a software project that has the ability to meet its predefined scope, the agreed software specifications in relation to usability and quality, and budget and schedule requirements, by following proper procedures, tools and techniques. Since there is no agreed definition of what constitutes project success, the three above-mentioned factors will be used to define software development project success in this research study.

Other than the three success metrics (i.e. time, cost and user specification) touted by many authors as being important for the success of a project, Wateridge (1998) has emphasized the need to consider all the other stakeholders involved in the development process. In fact, Wateridge (1998), Lehtinen *et al.* (2014) and Agarwal & Rathod (2006) are some of the few authors that combine internal characteristic of the project and external characteristic in their definition of software project success. While the category of internal characteristics refers to target time, cost and quality, the category of external characteristics refers to customer satisfaction and profitability. The definition of software project success needs to consider both characteristics. Agarwal & Rathod (2006) observed that while successful software projects are hard to define, the "*not successful software projects*", projects that are not delivered are even harder to define and measure.

Software projects can be delivered late or with increased budget, and the customer might be satisfied with its functionality. Although failure is not an option for a project manager, it occurs all the time and top-executives are often concerned about failure (Bergerm & Freund, 2012). Agarwal & Rathod (2006) generally found it difficult to judge whether a project is successful or not in situations where the project was delivered in the stipulated time and budget but with decreased project scope, or software that is delivered with desired scope and within timeframe but with high costs.

Projects fail for various reasons depending on the nature of project. A project to develop a particular software system (e.g. accounting system or payroll system) may be delivered on time, according to the desired specification and on budget, and satisfies the need of all interested parties; the same product may however not be profitable and other stakeholders will consequently categorise it as an unsuccessful project (Wateridge, 1998:60).

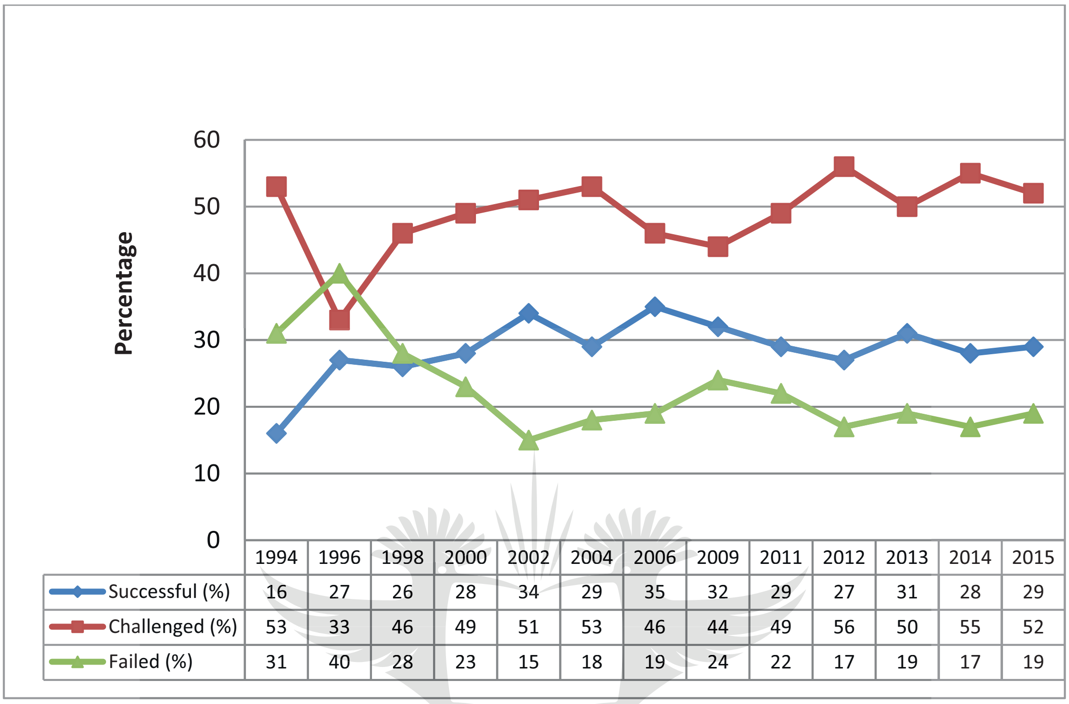
Lehtinen *et al.* (2014) defined software project failure as a recognizable failure to succeed in the cost, schedule and quality goals of the project. The term "*recognizable*" means that the identified contributors of the failed project can be avoided in future projects. According to Wateridge (1998), if the project does not meet time, budget and specification constraints, they assume the project has failed; however, projects can be classified as successful even if they were delivered late and over budget but have met the specifications. This means projects can be regarded as being successful even if they do not satisfy one or two success metrics. The most common definitions of '*successful project*' include three success metrics, namely: time, budget and scope. However, if one of the success metrics is not met, the project will not necessarily be regarded as a failed project. For example, clients can sometimes sacrifice one of the success metrics and accept the outcomes of the project even if the project is over the budget or has limited functions. If the project cannot meet all three success metrics, the project is definitely deemed as failure. According to KPMG (2013), executives should reward project managers with strong incentives for delivering a successful project, and also be held accountable for project failure.

The Standish Group (www.standishgroup.com) has a chronicle called the CHAOS Report, which has been reporting on the status and success rates of software development projects since 1994. The Project Resolution Benchmark is a comparative quality assessment tool that is used for measuring the success of their closed or completed projects against The Standish Group's CHAOS database. The Benchmark measures the six success metrics (individually and in combination) from the traditional and modern resolutions, namely: on time, on budget, on target, on goal, valuable, and customer satisfaction. Whereas the Traditional Resolution measures projects against the CHAOS database for on time, on budget, and on target (scope), the Modern Resolution measures projects against the CHAOS database for on time, on budget, with a satisfactory result. As far as the Modern Resolution is concerned, projects that were completed but are late, over budget and with unsatisfactory results are regarded as Challenged Projects. Projects that were cancelled or not used are deemed Failed Projects.

Challenged projects are generally regarded as projects that are between the success and failure scales, but have not been cancelled. Challenged projects have therefore partially failed, but have been delivered with limited scope and have overshot in terms of delivery time-frame or budget. The CHAOS Report defines a challenged project as a project that has been completed but not within budget and time and/or even lacks some of the functionalities that were expected. Whereas the stakeholders can view such as a project as a successful project, the same project can be viewed by the development team as a failure based on their own and different definition of '*project success*'.

Although the stakeholder can assess the outcome of the project on the basis of time and budget metrics, a developer can argue that the scope of the project determines the project outcome. The customer can also choose to ignore the time it has taken to deliver a software and the project cost that increased during the development of the software project provided the end-product satisfied their needs. The customer can also accept software with reduced functionality. According to Hughes, Ireland & West (2004), the final arbiter of the success or failure of a project is the project sponsor and the users of the delivered software product.

The CHAOS Report defines '*project success*' as projects completed on time and budget, with all features and functions as specified, otherwise project is considered failed when the project is cancelled prior to completion or not used after implementation or challenged if the project is over budget, late, and/or have unsatisfactory implementation. The IT software project performance data reported for the period 1994 to 2015 is summarised in **Figure 2.2**. Of the 50,000 software projects that were studied and analysed around the world in 2015, challenged projects were a regular feature at 52% (see **Figure 2.2**). The collated data used to produce **Figure 2.2** was sourced from the Hairul, Nasir & Sahibuddin (2011) and the Standish Group's 2015 CHAOS Report. The values displayed in **Figure 2.2** are based on all types of projects, and the outcomes of projects were summarised by taking into consideration the three success metrics, namely: on time, on budget with a satisfactory result. Therefore, this means that **Figure 2.2** has based the definition of '*project success*' on the Modern Resolution whereby all the software projects from 2011 to 2015 and on prior projects (from 1994 to 2010) are measured against the CHAOS database on three elements of success (i.e. on time, on budget, with a satisfactory result). The figures show that nothing has improved yet regarding project performance.



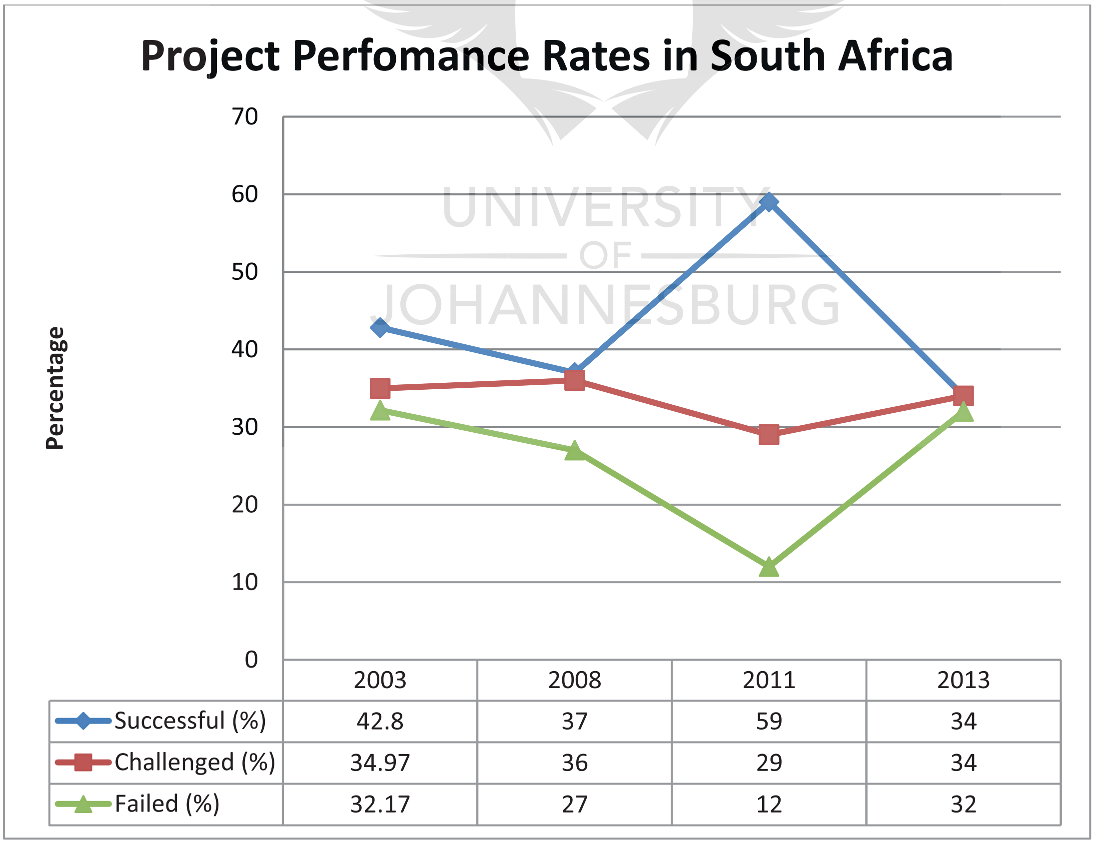
	1994	1996	1998	2000	2002	2004	2006	2009	2011	2012	2013	2014	2015
Successful (%)	16	27	26	28	34	29	35	32	29	27	31	28	29
Challenged (%)	53	33	46	49	51	53	46	44	49	56	50	55	52
Failed (%)	31	40	28	23	15	18	19	24	22	17	19	17	19

Figure 2. 2: *Standish IT software project performance rates from 1994 to 2015 (Adopted from Hairul, Nasir & Sahibuddin (2011), Hastie & Wojewoda (2015), and Marnewick (2013)*

Prabhakar (2008) has also alluded to a poor success rate of most projects reported in the media, which are either over budget or are late but are still regarded as successful. Figures reported on this category of projects was quite alarming for the periods 1994, 1996 and 1998. A slight improvement was recorded for the periods 1998-2002 and 2004-2006. During the period 2006 to 2015, the success rates decreased by 6% per annum.

As stated by (Jones, 2010), 'The software industry has the highest failure rate of any of the so-called engineering fields. According to Modern Resolution, in 2011, 22% of software projects were cancelled, while 49% were completed late and over budget. Only 29% of software projects were completed on budget and on time, also within a satisfactory result. In 2013, the cancellation rate figures improved by 2% from the 2012 figure of 17%. The four year figures from 2012 to 2015 showed an improvement; the average for the four year period is 18%, which is better than 2011. The success rates for the entire five year period (from 2009 to 2015) is almost the same and they range from 27% to 31%. The challenged projects maintained an average percentage of 52.5%. However, since challenged software projects were still in use and not cancelled, these projects might be considered partially successful since they perform other functions that they were developed for. Overall the failure rate of 17% to 20% is considered high.

A study of project success rate of IT projects in South Africa, which was initially undertaken by Sonnekus & Labuschagne (2003) was taken over by Marnewick (2013). The study reports on IT projects in general, and is not limited to software development projects. As shown in **Figure 2.3**, success rates of IT projects in South Africa in 2003 (43%) and 2008 (37%) were not as impressive as that of 2011 (59%). The results of 2011 by Marnewick (2013) almost doubled the results of 2008 (from 37% to 59%), the success rate of 59%, 29% were calculated as challenged, and failure rate was 12%. It must be conceded however that the results generated form the study of Marnewick (2013) were derived from projects that were undertaken in both South Africa and the continent at large. Results of the South African component of the study are presented in **Figure 2.3**.



	2003	2008	2011	2013
Successful (%)	42.8	37	59	34
Challenged (%)	34.97	36	29	34
Failed (%)	32.17	27	12	32

Figure 2. 3: *Project success rates in South Africa (2003-2013) (Joseph & Marnewick, 2014; Labuschagne et al., 2009; Marnewick, 2013; Sonnekus & Labuschagne, 2003)*

A comparative analysis of the results of project success published by The Standish Group (the CHAOS report) and those generated from the South African study has revealed that the locally generated performance rates are much better than those reported by The Standish Group. The result shows better performance of The Standish Report as 34%, which is the minimum result of The Prosperus Report. The Prosperus report results are always better than The Standish group results by huge margin. That means international software development teams are experiencing more challenges compared to software development teams in South African. The survey on organisations in the IT sector by KMPG (2013) has reported project success rate of 21% on key metrics of timely and delivery on budget and delivery on stated deliverables. These figures do not explain whether the drivers of such low success rate were triggered by project management maturity or not. There is a great need to measure performance rates of software development projects.

The attributes that contribute to the outcome of a software project are discussed in the next section. Specific factors that contribute to the success of a project such as effective communication with all the stakeholders and clear scope are very important (Hughes, Ireland and West, 2004). As part of a strategy to curb an increase in the failure rate of South African software development projects, Mnkandla and Marnewick (2011) have suggested that academic and professional training institutions should offer and apprenticeship in project management.

2.3 SUCCESS AND FAILURE FACTORS

There are many factors that contribute to the success of a project are known. Although the factors that contribute to the success of a software project are many, some have a minor contribution. According to Lehtinen *et al.* (2014:624), the common causes of software project failures include environment, tasks, methods and people. The high failure rates of software development projects are posing serious challenges to the software project industry and academics (Mtsweni, Horne & Van der Poll, 2016). As a result, Wateridge (1998:59) opined that: "*There is a need to identify how IS/IT projects, and their products are judged to be successful, and what factors are important in influencing that success*". Project success and project success factors need to be well defined to enable the project outcome to be fairly judged.

McLeod & MacDonell (2011) have monitored how factors that influence software development are perceived over a thirty-year period. The top ten factors that contribute to project success that have been reported by the Standish Group during the period 2009-2015 are listed in order of importance in **Table 2.1**.

Table 2. 1: *Success Factors (The standish Group, 2010-2015)*

Factors	2009	2010	2012	2013	2015
Executive support	2	1	1	1	1
User involvement	1	2	2	2	3
Clear Business Objectives	3	3	3	7	10
Emotional maturity	4	4	4	8	2
Optimization	5	5	5	3	4
Agile Process	6	6	6	6	7
Project management expertise	7	7	7	5	9
Skilled resources	8	8	8	4	5
Execution	9	9	9	9	8
Tools and infrastructure	10	10	10	10	
Modest Execution					6

Since 2010, executive support tops the list as the most important factor contributing towards the success of the project. The executive sponsor is the most important person involved with a project and should have the skills to lead and guide a project to resolution (The Standish Group, 2013). Also, it appears that successful projects have strong non-technical factors in terms of executive support and user involvement that may lead to clearly defined requirements and project objectives; technology, tools and methods play an important but less influential role (Marchewka, 2013:5). Projects that have an active executive sponsor have a higher rate of success. Similarly, lack of quality executive sponsor are likely to develop into Challenged and Failure Projects.

Other factors that are very important include user involvement, emotional maturity and clear business objectives. Unlike in civil engineering or construction projects or even in biochemical industry where the users are rare, user involvement in software projects is very important. In software projects, the user involvement is required from the beginning of the project until the project is in use or reached an operational stage. Software developers and system analysts have acknowledged that user involvement is one of the crucial factors to the success of software projects (Ruhe & Wohlin, 2016). As shown in **Table 2.1**, there has also been some movements for the reported period with respect to the Top 10 success factors. A new factor called Modest Execution, which was previously not included in the Top 10, has now replaced the Tools and Infrastructure factor. According to Chaos report (2015), modest execution is to have a process with few moving parts, and those parts are automated and streamlined, also means to use project management tools of very few features carefully. In 2015 project management expertise was rated factor number nine, even though the study by Erasmus et al. (2016) has established project management competency as the main factor that influences the success of software projects.

A cause analysis of software project failures by Lehtinen *et al.* (2014) revealed that there is no single cause of software project failures; the analysis also revealed that a lack of software testing plays a central role in the software project failure. The sequence of factors contributing to the success of software development project has not changed dramatically in the past 5 years. In the CHAOS Manifesto 2013 Report, the small software development projects that uses modern languages, methods and tools, for the period 2003 to 2012 was reported for each success factor. In addition to undertaking a comparative analysis of the different methods for software development project delivery under success factor Six (Agile), the special version of success factors for small projects was also presented in the same Report. Although the executive sponsor still topped the list as the most important factor contributing towards the success of projects, other factors shifted their positions on the list.

The Standish Group's CHAOS Reports seem to be widely recognised in the industry, because the Group's research started about 25 years ago and have reported on more than 80,000 completed IT and software development projects. Other similar studies appear to confirm their success factors. By looking into several pieces of literature in order to find more failure and success factors related to our study, we have found McManu and Wood-Harper (2007). In an attempt to look for failure and success factors related to our study, this study has identified a particularly important study by McManu & Wood-Harper (2007). Based on a research study of 214 European public and private projects, McManu & Wood-Harper (2007) separated the failure causal factors into management and technical causal factors (see **Table 2.2**). The management causal factors account for 65% of project failures, whereas technical causal factors account for 35%.

Table 2. 2: *Causal factors (McManu & Wood-Harper, 2007)*

Management causal factors	Technical causal factors
Poor leadership in project delivery	Inappropriate and ill-defined software requirements
Poor of stakeholder communication	Inappropriate technical designs
Poor competencies (and skill shortages)	Inappropriate development tool
Poor stakeholder management	Inappropriate user documentation
Poor estimation methods	Poor test planning
Poor risk management	Poor technical support
Insufficient management support	

As shown in **Table 2.2**, poor leadership in project delivery followed by poor stakeholder communication have been identified by McManu & Wood-Harper (2007) as the management causal factors with the highest contribution towards failure of software development projects.

Although the factors listed in **Table 2.2** do not differ much from the factors identified in the CHAOS Report, skills are categorised under each of the different top 10 success factors in the CHAOS report. The project success and failure factors are applicable to all projects environments, irrespective of whether a project is being conducted in a multi-project environment or an international setting (Camilleri, 2016).

Dorsey (2005) also mentioned three critical success factors that are common to all successful projects, namely:

1. Top management support;
2. A sound methodology; and
3. Solid technical leadership by someone who has successful completed a similar project.

The above critical factors need to be addressed as soon as possible so that they are not allowed to hinder a project to succeed at a later stage. Top management support, which is regarded by Dorsey (2005) as the main prominent success factor (2005), is also the biggest success contributor according to the Standish Group Report. As the leading contributor in the success of the project, top project management support must therefore be addressed as soon as the project commences. Also, an appropriate methodology such as PRINCE2 must be selected and implemented.

Kaur & Sengupta (2013) are of the opinion that the following factors make a significant contribution towards the failure of software development projects:

1. Project team compromised;
2. Inability to handle varying demands from clients;
3. Estimation misjudgement;
4. Unclear goals; and
5. Change of management during development.

If the organisational and development teams are aware of the factors that contribute to the failure of the projects it becomes easier to drive the project in the right direction. Some of the factors listed are already addressed by many researchers including the implementation of project management maturity model as other factor that contribute to the project performance. Hairul, Nasir & Sahibuddin (2011) have identified 26 other success factors that contribute to the success of software development project. These 26 factors were identified following a scan of 43 articles. In general, some of the factors are common across all projects. Attarzadeh & Ow (2008) have listed lack of IT management as a project impaired factors on his study. The impaired projects are project which are cancelled at some point during the life cycle of the project.

Factors behind the outcome results of a software project are reported each and every year. A review and clarification of these factors can potentially contribute to or lead to a successful production of the desired software product. Many organisations start to develop their maturity by addressing success factors in the project environment (Association for Project Management., 2012). The success factors provided by The Standish Group Report are not the same as factors reported by other researchers such as Marnewick (2013). The regular top four success factors reported by The Standish Group Report are: executive support, user involvement, clear business objectives and emotional maturity. In a study involving the status of ICT Project Management within African countries and some software development projects, Joseph & Marnewick (2014) have reported that their top 10 most influential success factors (e.g. common attributes such as executive support, user involvement and clear objectives) are almost the same as those reported by the Standish Group. As shown in **Table 2.3**, the top three success factors listed by Joseph & Marnewick (2014) are completely different from those of the Standish Group (CHAOS Report). The results of CHAOS Report are based on a longitudinal study which is collecting data over a period of time, while the study by Joseph & Marnewick (2014) was a cross-sectional study which happens once.

Table 2. 3: *Comparison of Top 3 Success Factors (extracted from Joseph & Marnewick, 2014; The Standish Group)*

CHAOS Report success factors (2010 to 2015)	ICT factors by Joseph & Marnewick (2014)
1.Executive management support	1. Requirements definition clarity
2.User Involvement	2. Communication between team and customers
3.Clear Business Objectives	3. Communication between project team members

The list of factors provided by Kaur & Sengupta (2013) are also based on software development projects and are related to the factors provided by the Standish Group. The factor that is common to both studies is clear business objectives, although Kaur & Sengupta (2013) used the word unclear goals to emphasize failure cause. Dorsey (2005) has listed top management support as the number one factor; this factor is referred to as the executive support in the Standish Group Report. This means that the executive management support is more important and it is an essential component required for projects to succeed. McManu & Wood-Harper (2007) listed the following three other factors that are also found in top ten list of the Standish Group Report: Skills shortage, management support and development tools.

It is clear from the literature scan undertaken in this section that the success factors that are reported by most researchers have also been reported in the Standish Group Report with minor differences relating to the ranking of these factors. Mtsweni et al. (2016) has identified and categorised soft skills as a factor that influences the success of a software development project. Of the 20 known factors, many researchers have mentioned as few as three or five factors. The study by McManu & Wood-Harper (2007) does not cover all factors reported by The Standish Group Report. A maturity model addresses the performance factors of the projects.

2.4 PROJECT SUCCESS AND MATURITY

Organisations need to measure their maturity level against an industry standard such as CMMI from Software Engineering Institute (SEI) (Kahate, 2004). The CMMI is always available and accessible to assist organisations with a consistent approach to software development and engineering processes (Jacobs, 2011). CMMI is a maturity model used by many organisations all over the world. When a company follows a maturity model, the outcome of a project can be realized as early as possible (Kerzner, 2013). The issue of software project failure had been an ongoing challenge for a long period of time. The software industry has maturity models to assist in the assessment and improvement of both the maturity levels of organisation producing software as well as software process capability (Peldzius & Ragaisis, 2011; Spalek, 2013).

Generally, maturity means fully developed or perfect. Most maturity models have 5 levels, ranging from 0 or 1 to 4 or 5, respectively (Hwang, 2009; Kaur, 2014; Kwak & Ibbs, 2002; Niazi, Wilson & Zowghi, 2005 ; Marnewick & Ramachandran, 2009; Paulk, 1993, 1995; Paulk *et al.*, 2003). A high maturity organisation is an organisation that operate at maturity level four (Kulpa & Johnson, 2008). To move a large organization from lower levels to upper levels of maturity takes several years. If you try to move from level one to level five across your whole organization in one step, you are bound to fail (Humble & Russell, 2009).

The first level is associated with a low level of maturity, and upper levels are associated with more maturity. Cooke-Davies & Arzymanow (2003) discovered that more matured organisations are those that have adopted a maturity model long before those that have adopted the model recently. Nazar and Abbasi (2008) reported that matured organisations have benefited from better performance after reaching a particular maturity level. This means organisations can realise benefits such as cost savings and faster time to market when they reach higher maturity level as compared to low level maturity organisations ('Project Management Maturity & Value Benchmark 2014', 2014). Based on a recent study by Klaus-Rosińska & Kuchta (2017), the benefits of fully matured organisations are on-time project delivery, organisational profit, ability of the organisation to reduce costs and organisational efficiency. A matured organisation is able to achieve the goals it sets for itself. Silva *et al.* (2015) have asserted that: "An organization that offers a rating at the highest levels of these maturity models excels in competitions for software projects". Each maturity level is composed of several key process areas, besides level 1. Other major component of the maturity model is its capability levels. The capability refers to the ability of the organisation to produce and provide products consistently and predictably.

Maturity means an organisation has the knowledge and potential to grow in capability of their industry area (Donaldson and Siegel, 1997), and indicate level of experience of organisation when it comes to projects. Paulk *et al.* (2003) and Chrissis *et al.* (2011) defines maturity level as a well-defined evolutionary plateau towards achieving a mature software process. Maturity model is a framework for improving the ability to manage projects. Maturity in project management organizations is when processes are in place and followed to deliver projects, and it presents a good chance that each project is likely to succeed (Kerzner, 2013:34). Maturity models are regarded as frameworks that can transform an organisation from being less organised, less standardised and less documented into an organisation that can achieve higher standards and greater consistency (Ofori and Deffor, 2013). The word project maturity might indicate or even measure an organisation's ability to use and manipulate projects for other purposes (Andersen and Jessen, 2003). Maturity models are a means to measure the rate of capability of an organisation. Proença & Borbinha (2016) have defined a maturity model as a proven technique that is valuable and reliable in measuring aspects of a process or an organisation.

Most of the time, an organisation's level of project management maturity will influence the outcome of that organisation's project (Project Management Institute, 2013:19). There is enough evidence in the literature to indicate that organisations with higher maturity levels are expected to be successful when dealing with project effectiveness and efficiency, and must have a competitive advantage in the market place (Backlund, Chronéer and Sundqvist, 2014). Silva *et al.* (2015) supported the idea by suggesting that an organisation that offers a rating at the highest levels of these maturity models excels in competitions for software projects. Organisations that are rated highly in maturity deliver software on time, meet budget and increase profit (Kelsey, 2006). When a company has attained a higher level of project management maturity, the project management costs are generally lower than what it costs for less mature peer organisations (Ibbs and Reginato, 2002). Since maturity models were developed to help organisations and software teams to deliver projects consistently, within budget and on time, the benefits should be realised. The higher the level of maturity, the better the software development process (Kerzner, 2013). The organisations that are rated highly on maturity level should deliver projects successfully and easily than lowly rated organisations. A matured organisation should deliver successful project with minimum effort. Although it takes time to adopt a maturity model, once an organisation reaches higher levels of Project

Management Maturity (PMM), Project Management (PM) costs start to decrease (Ibbs and Reginato, 2002).

To ensure organisational success in the global business environment, it is required that organisations ensure a higher standard of performance (Ofori & Deffor, 2013). Organisations that complete projects successfully receive credits and recognition. The assessment of the project management maturity level by a company, shows how the company is committed to manage projects (Spalek, 2013).

Software project performance is regularly measured in higher-maturity organizations (Kelsey, 2006). Other studies determine the relationship between PMM and project success. Jiang *et al.* (2004) has concluded that a CMMI organisation develops higher quality software and increases project chances to success. Data obtained from measuring completed software projects could be used to improve new software projects by software development teams. The ability to produce a quality software project and successfully should depend on the maturity level of the processes used to build the software product. There is however no strong evidence in the literature to support this idea. For this reason, Gomes, Romão & Carvalho (2016) contend that maturity models emerged as roadmaps for strategic improvement, not a proven map to project success.

Software project performance is measured for variety of reasons, maturity model can help the development team to survive the affected project, and the software development processes must be controlled. A member of higher maturity team must have the ability not to fear a failure because they will use the skills earned from failed project to deliver a new project successfully. Quality developed software is fundamental to the success of the software development industry.

Jiang *et al.* (2004) has suggested that it may take organisations some years to achieve the next level of maturity and realising the benefits; achieving higher levels of maturity is basically a long-term commitment. Spalek (2013) has found that an increasing level of maturity in project management can have an influence on the reduction of costs of projects managed by the company. The organisations that implement a maturity model must be patient to rip the benefits.

Following a survey on IT project management, Zarrella (2005) emphasized the need for higher maturity levels of project management for the organisation to survive the global software project competition. The adoption of formal methodologies has a higher impact on the success of the software project, although the contribution of maturity to the direct success rate of South African projects needs to be studied.

PM maturity is one of the major contributors to project success; it can apparently become a way of ‘doing businesses’ for organisations resulting in bigger market share (Mittermaier & Steyn, 2009). Many researchers have realised that maturity is associated more with project performance than success (Jiang *et al.,* 2004). According to KPMG (2013), research project management maturity correlates highly with success.

A study of PriceWaterHouse Coopers (2004) supports the notion project management maturity has on project performance. Pennypacker *et al.* (2003) relates higher maturity level and the project performance by reporting that 30% of mature organisations showed over 25% improvements when compared with organisations rated on lower maturity levels. A PWC (2004, 2007 and 2012) survey has revealed that a higher maturity level goes hand-in-hand with a higher project performance level. Every project team or project organisation wants to obtain consistent results on their projects and the project outcomes of organisations without a project management maturity model in place are dependent on their ‘star’ developer. In an immature organisation, when the star developers or lead developer resigns, their projects suffers (Paulk, 1995). Project teams that have a maturity model in place depend on the project successes and not on an individual’s experience because the selected maturity model will guide the team to deliver their project. Peldzius & Ragaisis (2011) understood that software process maturity is not isolated but related to project success and quality of software product.

Following a project management methodology such as PRINCE II does not guarantee a project will be successful; however, if applied carefully it will provide management with the means to be successful (Hughes, Ireland and West, 2004). The study by PricewaterhouseCoopers (2004) has revealed that PMM maturity correlates highly with project success. However, Labuschagne, Jakovljevic & Marnewick (2009) have found no significant correlation between project success and maturity level of an IT organisation in South Africa. PwC is among the few research institutions that found a direct link between maturity levels and project performances, which is one of the objectives of the study that need investigation. According to Pretorius *et al.* (2012), there is no statistical figures to support the existence of a positive correlation between maturity level and performance. Although very few publications are available in the literature that acknowledge the lack of evidence supporting statistical correlation between maturity level and project success, organisations with higher maturity levels are nevertheless still expected to complete projects successfully. The successful adoption of maturity model by development team will contribute to the success of software projects. Organisations that improve their process maturity stand to gain the following benefits:

- Improved quality (Chrissis *et al.*, 2011:8; Jones, 2010:324)
- Improved schedule and budget predictability (Warrilow, 2009; Bourne, 2011)
- Improved productivity (Chrissis *et al.*, 2011:8; Jones, 2010:324; Bourne, 2011)
- Increased customer satisfaction (Hairul, Nasir & Sahibuddin, 2011; Bourne, 2011)
- Improved employee morale (Jones, 2015; Bourne, 2011)
- Measurement of project performance (Warrilow, 2009)
- Decreased cost of quality (Bourne, 2011)
- Implementation of software process improvement (Niazi, Wilson and Zowghi, 2005)

Organisations on maturity levels 4 to 5 have realised the improved projects results (Humble & Russell, 2009; 'Project Management Maturity & Value Benchmark 2014', 2014). Organisation that have achieved a higher maturity level, benefits more than those organisations that are rated very low on maturity because such organisations can deliver projects or portfolios with more efficiency. An empirical study on project management by Spalek (2013) reported that 70% of IT companies are on maturity level 3. PWC (2015) used the PWC maturity model and found that 62% of organizations surveyed were operating projects within the level 4 or 5 of maturity. Furthermore, Ofori & Deffor (2013) found that non-profit organisations exhibited higher levels of maturity levels when compared with the other categorised phases.

The organization's level of PMM and its project management systems can influence the project outcome (Project Management Institute, 2013). Many standardisation organisations such as PMI, Government Commerce (OGC), Australian Institute of Project Management (AIPM) and CMMI have developed their own assessment certifications for individuals and organizations that are interested in project management practice. Researchers and standardisation organisations have also designed different project management maturity models (PMMMs) to evaluate PMM for organization (Farrokh and Mansur, 2013). Any type of organisation such as companies that offer business related projects can utilize PMMM for the measurement and improvement of their project management competence (Albrecht & Spang, 2011). The study of Marnewick (2013) is one of the studies that has found that the relationship between project success and maturity levels, albeit a weak relationship, is significant. Maturity levels can be inticated by bar charts or histogram.

2.5 PROJECT MANAGEMENT MATURITY

Bay & Skitmore (2006) defines *project management* as "a general purpose management tool that can bring projects to a successful completion and to the satisfaction of the project stakeholders, given the traditional constraints, of defined scope, desired quality, budgeted cost, and a schedule deadline". Among all the attributes used to express the definition of project management by Bay & Skitmore (2006), the definition of the concept is centred around the phrase '*successful completion*'. Some of the software problems such as cost and schedule overruns were however attributed to poor project management. For software development to succeed, project management maturity of the organisation needs to be high.

Since the mid-90s, a couple of project management maturity models have transpired (Klaus-Rosińska & Kuchta, 2017; Pennypacker & Grant, 2003; Pretorius *et al.*, 2012). Project management maturity is an ongoing development of an organisation strategy to project management approach, which caters for methodology and core decision processes (Ofori & Deffor, 2013). Project management maturity supports an organisation from the day it implements a model. Organisations can develop their own project management maturity models based on existing models and their needs. Project management maturity models provide a roadmap that directs or shows an organisation how to move from immature level to more matured levels of project performance. Project management maturity is a complex-measuring tool used by an organization to measure its current project management standard and processes (Kwak & Ibbs, 2002:150; Ibbs, Reginato & Kwak, 2004:1216). According to Judgev & Thomas (2002), project management maturity models could be an answer or a support system to link projects with strategy and organization. Maturity models are developed with a common purpose; to improve the maturity level of the organisations that use them and to improve project processes.

There are many existing various maturity models such as CMMI, OPM3, PWC's PM Maturity model, Kerzner's PMMM and P3M3 (Khoshgoftar & Osman, 2009), Berkeley Model, IPMA-Delta (Archibald & Prad, 2014) are known. Although this does not form the scope of this discussion, it is important to note that PwC's PM Maturity Model uses different names for labelling the maturity levels. Some of popular maturity models and their attributes that will be discussed in this section are listed in **Table 2.4**.

Table 2. 4: *Attributes of popular Maturity Models*

Maturity Levels	OPM3 (PMI)	P3M3 (OGC)	PMMM (Kerzner)
Level 1	Standardize	Awareness	Common Language
Level 2	Measure	Repeatable	Common Processes
Level 3	Control	Defined	Singular Methodology
Level 4	Continuous improvement	Managed	Benchmarking
Level 5		Optimized	Continuous Improvement

All the maturity models have advantages and disadvantages. Many of the project management maturity models are complex and therefore not usable at all. In contrast, other PMMMs are simple, straightforward and easy to use. The following are regarded by Nenni *et al.* (2014) as the top 3 maturity models:

1. Organizational Project Management Maturity Model (OPM3)
2. Project, Program, Portfolio Management Maturity Model (P3M3)
3. Kerzner's Project Management Maturity Model (K-PMMM)

These maturity models are unique in terms of their characteristics and there are no guidelines for selecting one of the models to use. Maturity models can be applied by project team or to a department or business unit with a desire to improve the way their projects are being managed.

Each maturity level consists of "key process areas (KPAs)" that are defined by "key practices". KPAs specify the issues that need to be addressed by maturity model first in order to achieve a specific maturity level. KPAs are a group of related activity that are defined to reside in a single maturity level (Donaldson & Siegel, 1997).

For the purpose of this dissertation, a comparative analysis of the maturity models listed by Nenni *et al.* (2014) is undertaken in the sub-sections that follows. Khoshgoftar & Osman (2009) have acknowledged that OPM3 is more suitable for managing projects when compared with other maturity models. It is on this basis that this maturity model is discussed first.

2.5.1. THE ORGANISATIONAL PROJECT MANAGEMENT MATURITY MODEL (OPM3)

OPM3 is a maturity model that covers the needs of different organisations, and was it designed in such a way that it is very easy to follow and understand (Kalus-Rosinska & Kuchta, 2017).

The OPM3 model was originally developed by volunteers led by John Schlichter, the founder of OPM Experts. OPM3 focuses on integrating three domains (Project, Portfolio and Program) and defines 4 maturity stages (standardize measure, control and continuously improve). The purpose of OPM3 is to guide organisations from the lowest level of maturity to the highest level in project, portfolio and program. The organisation can address one domain at the time, combine any two that suits their needs, or address all of them simultaneously. OPM3 offers most comprehensive assessment and reporting supported by software. The OPM3 model was mostly used in the construction industry and as a basis for the development of a new maturity model (De Souza & Gomes, 2015).

OPM3 is based on widely accepted Project Management Body of Knowledge (PMBoK), the program management project and portfolio management tool that it is capable of assessing organizational maturity at any level. OPM3 is modular and scalable; organisations can implement only those parts that are most relevant to the environment. The OPM3 maturity model is illustrated in **Figure 2.4**. The project management processes in each process group within domains are achieved by a logical path of improvement of standardize, measure, control, and continuously improve (OPM3, 2013).

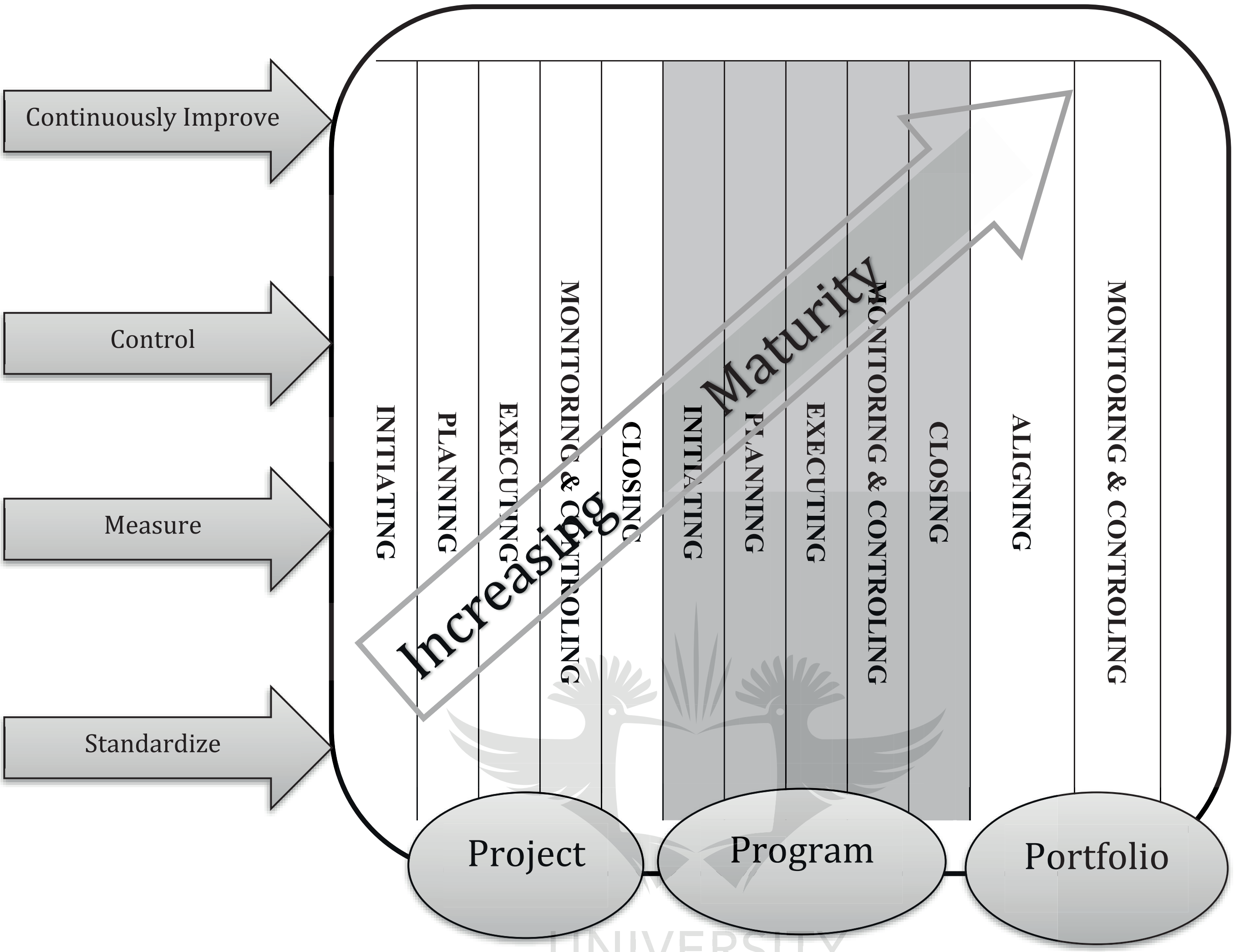


Figure 2. 4: *An illustration of the OPM3 maturity model (Adapted from OPM3, 2013)*

The maturity of projects, programs and portfolios depends on the progression of maturity stages across five process groups. OPM3 focus more on best practices and capabilities. The organisation that poses high maturity level on portfolio practices, does not mean they perform well on program or project management. Organisations may perform well when it comes to project management but perform very badly when it comes to portfolio management. The OPM3 maturity model has 151 questions that organisations or individuals can use to assess the state of maturity level and compare with other best standards.

The industry leaders have implemented OPM3 to transform their ability to close the gap between strategic target goal and tactical outcomes (OPM Experts LLC, 2016). Some of the benefits of OPM3 listed by OPM Experts LLC include:

- Greater capability to deliver projects successfully, predictably and consistently.

- Increased productivity
- Improved decision-making
- Greater capability to choose the right projects
- Better performance data for executives

According to Khoshgoftar & Osman (2009), OPM3 has unlimited benefits in the industry and is the best maturity model that improves organizational performance.

2.5.2. PORTFOLIO, PROGRAMME AND PROJECT MANAGEMENT MATURITY MODEL (P3M3)

The portfolio, programme and project management maturity model (P3M3) was first developed in 2006 by the Office of Government Commerce in the United Kingdom (UK). The P3M3 model was developed based on interest to bridge the gap between organisational strategy and successful projects. Like the CMMI, the P3M3 maturity level is known by the five levels of maturity frameworks listed in **Table 2.5** (Warrilow, 2009; Silvius et al., 2012; Tahri & Drissi-Kaitouni, 2015):

Table 2. 5: *Levels of maturity frameworks of P3M3*

I	Awareness	Most of the organisations run both projects and programmes with no planning and lack of control at awareness level. The company can deliver projects without a standard process, but there is a high possibility of project delays. Other organisations that are at this initial level depend on experience of individuals to survive and deliver projects.
II	Repeatable	At this level, an organisation has a knowledge of PM, but the PM has no standardisation platform, which lead to project failure. The planning and control of projects and/or programs are still isolated. The practice of project management is at the beginning.
III	Defined	Now the organisation has adopted the standards that can be used to direct a project or programme. Also, the data or information of

		previous project results is available, which can be used to avoid delays and cost over-runs.
IIII	Managed	The organisation reduces any obstructions towards positive project outcomes, and actively improves skills of team members: e.g. negotiation and conflict related skills. The organisation must predict outcomes better.
V	Optimised	The organisation finally shows that it has reached a higher level of project management practice and discovery. The feature of optimisation is a higher level of success. The results in terms of factors such as cost, time and quality are optimized.

The model is aligned with the PMBoK. The P3M3 explains the portfolio, programme and project related activities with the key processes areas that influence the project outcome (P3M3, 2006). The maturity model allows an independent assessment of project, programme and portfolio (Young *et al.*, 2014). P3M3 was derived from the CMMI model and it has three different sub-maturity models, which are:

- PfM3_Portfolio Management Maturity Model
- PgM3_Programme Management Maturity Model
- PjM3_Project Management Maturity Model.

P3M3 encourages independent assessment between its maturity models, which means organisation may be better on project management than it is on portfolio management, or even better on two models. P3M3 quantifies organization's performance to the following seven key management perspectives across all the three models that can be evaluated at all five levels of maturity (Silvius *et al.*, 2017; Young *et al.*, 2014:219):

- Organisational governance
- Management control
- Benefits management
- Risk management
- Stakeholder management

- Finance management
- Resource management.

Based on the maturity level an organisation exhibits in any of the three sub-maturity models of P3M3, the organisation obtains results of its performance; the procedure for maturity assessment will follow should a need for improvement arise. **Figure 2.5** shows the alignment of 3-sub maturity model against the seven key areas mentioned above.

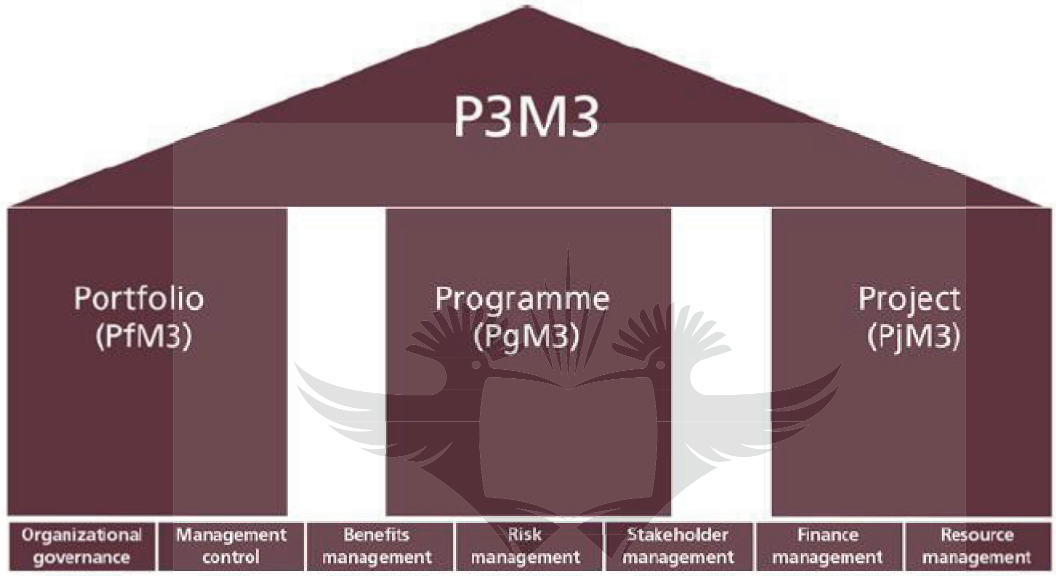


Figure 2. 5: *Project, Program and Portfolio Management Maturity Model (OGC, 2011)*

P3M3 is a kind of project management maturity model that covers 42 KPA/KPI. An organisation that has achieved P3M3 maturity levels 4-5, can achieve different maturity levels when tested by different maturity models. According to KPMG (2013), it takes on average 18 months to move up a level in the P3M3. Similar to other maturity models, the benefits of an organisation using P3M3 are high rate of return on investment (ROI); lower costs; better customer satisfaction; boost employee morale and better quality of overall projects (Young *et al.*, 2014).

2.5.3. KERZNER'S PROJECT MANAGEMENT MATURITY MODEL (K-PMMM)

The achievement of quality in PM practice depends on the K-PMMM. K-PMMM describes five levels that constitute this model. K-PMMM portrays itself as an alternative model to the well-known model called CMMI. The model uses different approaches to assess maturity levels. Each of the five levels, which are describe below, denote a different type of maturity in

PM as prescribed by Kerzner (2013).

I. *Level 1: Common Language:* At this level, the organization may not consider itself maturing; this level means the organisation lacks maturity and does not have the necessary knowledge to establish a process.

II. *Level 2: Common Processes:* Essential processes are defined in this level; the organization is now knowledgeable about working with processes. The process can be implemented and followed successfully. Common processes are defined and developed to influence the success of the projects. Different methodologies can be employed and tested in this level.

III. *Level 3: Singular Methodology:* Starting from this level and above, the focus is to mature one process at the time. Singular methodology layer is developed by combining related methodologies under project management. Organisation now maintain and update processes regularly easier under single methodology.

IIII. *Level 4: Benchmarking:* In his level, an organisation realizes the importance of process improvements. For the organisation to remain competitive on services, it must benchmark continuously.

V. *Level 5: Continuous Improvement:* This stage come after benchmarking. In this level, an organisation can collaborate with customers on its customer process. The organisation has reached the stage of reviewing information gained through benchmarking and react to changes and possible competition events.

K-PMMM differs from OPM3 and P3M3 because it covers strategic project management only, portfolio and program management are not covered. As shown in **Figure 2.6**, the maturity model of Kerzner emphases that initial levels must be completed before levels that follow can be completed.

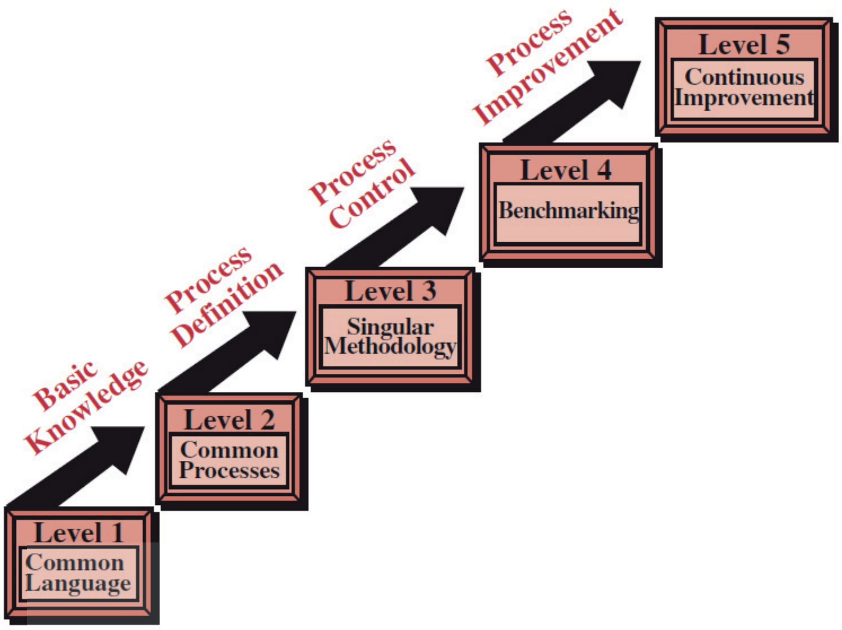


Figure 2. 6: *The five levels of KPMMM (Kerzner, 2013:737)*

When K-PMMM is adopted, the organisation can only struggle at maturity level 3, which has higher risk and degree of difficulty for the project team and organisation. The degree of difficulty to archive level 1 and 2 is medium. To achieve level 3 requires extra effort and commitment to shift in the corporate culture. If an organization has achieved level 3, the time and effort required to achieve maturity levels 4 and 5 will have a low degree of difficulty (Kerzner, 2013:740). The benefits of the K-PMMM maturity model might only be realised when the organisation has achieved level 3 and above (Jiang *et al.,* 2004). The highest level of maturity is level 5, meaning an organisation can continuously search for higher improvements of its project management processes (Spalek, 2013). The degree of difficulty to move from an immature level to more mature levels is illustrated in **Figure 2.7**.

Level	Description	Degree of Difficulty
1	Common Language	Medium
2	Common Processes	Medium
3	Singular Methodology	High
4	Benchmarking	Low
5	Continuous Improvement	Low

Figure 2. 7: *Degree of effort required on each of the five levels of maturity (Kerzner, 2013:740)*

Each stage of the five levels denotes a different kind of maturity in PM. The criteria of Khoshgoftar & Osman (2009) was used to make the comparisons of 3 shortlisted PMMs (see **Table 2.6**).

Table 2. 6: *Project Management Maturity Models (Khoshgoftar and Osman, 2009)*

Criterion	**OPM3**	**P3M3**	**K-PMMM**
Publisher	PMI	OGC	ILL
Structure	Multi-Dimensional	Staged	Staged
Maturity level	1-4	1-5	1-5
Refer to standard	PMBOK	MSP	PMBOK

OPM3 is the only PMM that is multi-dimensional; other maturity models are staged only. K-PMMM is comprised of 3 to 6 levels, although most models have 5 levels.

The analysis of the existing maturity models by Neverauskas & Railaite (2013) have shown that some models are based on the 10 project management knowledge areas, and not only are some models concentrated on the project, like P3M3 they are also focussed on the program or project portfolio management. OPM3 is the only maturity model that does not feature five levels.

Project management maturity and software development maturity are not the same thing. The following section will give a detailed account of software development maturity, and the section that immediately follows will describe the relationship between project management maturity and software development maturity. There are very few well known software development maturity models as compared to project management maturity models. Software development maturity models are not multi-dimensional like OPM3, they only evaluate project management capability of the organisation.

2.6 SOFTWARE DEVELOPMENT MATURITY

In order to improve software development or the process that produces software, a standard measurement tool is required. Measurement tools helps organisations to measure progress, increase value, reduce costs and promote on time project delivery (PricewaterhouseCooper, 2013). Software development performance is regularly measured in high-maturity organisations (Jalote, 2002). Developers are changing the world of IT by coding software, and developers can influence and recommend any software development maturity to their employers. According to Larrucea *et al.* (2016) and Niazi, Wilson & Zowghi (2005), larger organisations typically have used traditional Software Process Improvement (SPI) models such as CMMI and ISO/IEC 15504, and smaller organisations regard software process models as standards for bigger organisations and they, as a result do not employ or adopt maturity models. According to Jones (2010), larger software organisations are those that have more than 1000 software workers, and smaller organisations are those that have employed fewer than 25 employees or less. The leading maturity models in software development are models such as ISO/IEC 15504 and CMMI-DEV, they emphasise the need to manage, establish, measure and optimise processes (Fontana *et al.*, 2014). Organisations that develop software product using software maturity models such as ISO/IEC 15504 and CMMI-DEV are guided by defined detailed process-oriented maturity (Fontana *et al.*, 2014).

Cheema & Shahid (2005) have found that the software industry and software companies recommend their own maturity models. Most companies have developed their own maturity models. For example, Systems, Applications and Products (SAP) developed a SAP maturity model that has 5 tiers like most maturity models, and PwC has developed its own model called PwC maturity model that consists of five levels.

Software development teams adopt software maturity models such as CMMI-Dev for variety of reasons; others adopt it so that they can build their own maturity model. Software development maturity models advances the practice of software development. Software projects are more complex than other types of projects.

There are maturity models that are service oriented, and other maturity models are for software testing and implementation. Organisations should strive to increase quality of their software projects. Software development efforts and methodologies used needs to be linked to the overall goals and objectives of the organization in order to maximize the level of project success.

The systematic literature review of capability and maturity models by Von Wangenheim *et al.*, (2010) has identified CMM (SW-CMM), CMMI/CMMI-DEV, ISO/IEC 15504, ISO 9000 and ISO/IEC 12207 models as the most used models. Patel & Ramachandran. (2009) acknowledged that CMMI has gained a lot of attention and popularity, but for agile software development environments, he prefers the Agile Maturity Model (AMM). CMMI and ISO's SPICE

(ISO/IEC 15504) were recommended by Niazi, Wilson & Zowghi (2005). For the purpose of the study, only three of the maturity models (i.e. CMMI-DEV, ISO/IEC 15504 and AMM) will be discussed and compared. The main purpose of software development maturity reporting is to provide clients with useful information before clients can chose the provider and before money is spent; companies must also adopt proven best practices and avoid harmful practices.

Standards such as CMM, CMMI, ISO, and IEEE have introduced comprehensive quality assurance activities in the software life cycle, and the waterfall model has as a result incorporated many of these aspects (Ahmed, 2012). Enough evidence exist to prove that organisations that use a quality management system driven by the CMMI model or the IEEE standards have a better chance at constraining defect injections and determining residual risk (Kelsey, 2006:45). These standards define the steps that must be followed during the planning and execution of the software development projects. IEEE standards are focused on using standard processes and tools to achieve project excellence and make software projects more successful (Ahmed, 2012). IEEE and ISO standards focuses more at project level, and CMMI is at organisational level. CMMI-DEV is a model that provides best practices in development environments; it is not only for software development, it is also for any product development. CMMI-DEV has 5 maturity levels, and each level is built on the previous level. The process areas and the practices are not only related to software development; they cover the process management aspects of the organisation as well.

CMMI-DEV covers practices that include process management, hardware engineering, systems engineering, project management and supporting processes used in the maintenance, services and development phases. CMMI for development is discussed in more detail in the following section. Thereafter, other software maturity models are discussed. Not enough information is currently available on Agile Maturity and ISO/ICE 15504, and these standards will therefore not be discussed to the same extent as CMM-DEV.

2.6.1. CMMI DEV

Capability Maturity Model Integration (CMMI) is an industry framework leader that improves product quality and development efficiency for software and hardware (Team, 2010). CMMI can guide software and hardware development from the beginning of project throughout deployment and maintenance phases. CMMI is freely available; any organisation can download it and test it. The CMMI standard is declared as a *de-facto* standard for the software industry (Mutafelija a& Stromberg, 2003; Team, 2010). This means any organisation without maturity a model in place to improve their projects can adopt it for any kind of project. The main purpose of CMMI for development is to assist organisations with development and maintenance processes of products and services (Pino *et al.*, 2010).

The CMMI model covers five maturity platforms through the processes that evolve from initial to manage, quantitatively manage and, then finally optimize (Fontana *et al.*, 2014; Cooke-davies & Arzymanow, 2003). CMMI evolution ranks software development teams and organisations into one of the five levels. The experience gained from executing projects will help software development teams or development organisations to keep on maturing. The benefits realisation of moving from low maturity level to higher maturity level is when the development teams start observing the results of improved software production.

The main objective of the CMMI is to offer direction and regulations for improving an organisation's processes and the aptitude to survive the challenges of development, acquisition, and maintenance of products (Von Wangenheim *et al.*, 2010). To this end, the three-current assemblage of the CMMI framework are CMMI for Development (CMMI-DEV), CMMI for Services (CMMI-SVC) and CMMI for Acquisition (CMMI-ACQ) (CMMI SEI, Carnegie Mellon). CMMI framework can help any organisation to improve performance on its industry. The benefits of CMMI to an organisation include identifying and solving problems, and this covers quite a large number of activities. The choice of a model for a particular organisation is dependent on the type of the business and business objectives of that particular organisation. Since the focus of this study is on software development, only CMMI-DEV will be discussed. The main objective of CMMI-DEV is to assist software organisations of various sizes (i.e. small to medium and large) to improve their development processes.

CMMI-DEV has two improvement paths, namely staged and continues, which an organisational unit can choose to implement: (Fontana *et al.*, 2014; Persse, 2007). The staged representation model, which was developed with the original version of CMM (Persse, 2007), is compatible with the software capability maturity model (CMM) that was proposed in the late 1980s for assessing the maturity of an organisation (Tsui, Karam & Bernal, 2014:73; Sommerville, 2011:727). The staged representation model defines process areas for the area that needs improvement and progression to higher maturity levels; this is then followed by continuous representation that improves process areas individually and to propulsion to a higher capability level. This means that when the project teams or development organisations follow the continuous representation, they achieve capability levels not maturity levels. The continuous representation enables the selection of one or more process areas for improvement (Mutafelija & Stromberg, 2008). As to which process areas to implement will depend on the needs within an organisation. Following the staged path means that the organisation is grouping process areas according to the maturity levels. In staged representation, any level between the five maturity levels can be achieved. In short, the staged representation focuses on advancing maturity, and then continuous representation is on capability. The staged representation measures organisation maturity based on five level scale ranging from maturity level 1 to 5 and the organisation has no option to select which process areas to adopt. If an organisation that adopts the staged representation has the objective of reaching any maturity level (e.g. 3 or 1 or any level), it must adopt all process areas prescribed for that level (Persse, 2007). Continuous representation on the other hand measures organisational capability on a six level capability scale from zero to five. The difference of naming the levels is on capability level 0-1 and maturity level 1. Capability level 0 = Incomplete and 1 = Performed, while maturity level 1 = Initial.

CMMI-DEV is a collection of best practices that address requirements development and maintenance activities that product life-cycle goes from the start of the project until maintenance (Tamura, 2009; Von Wangenheim *et al.,* 2010; Zhang & Li, n.d.). Originally the total number of key process areas was 18 during that time of CMM (referred to as *KPAs*), these are now replaced by 25 process areas (PAs), which have been reduced further to 22 process areas. The process areas are grouped by the capability of maturity level and the organization is expected to success regularly (Galin, 2004). The latest model, which is CMMI-DEV 1.3, consists of 22 process areas classified into four classifications: Project and Process Management, Engineering and Support. The engineering category deals with issues such as how to design, test and code (Jalote, 2002), and the project management categories focus on planning and controlling the engineering activities so that the software can meet project goals.

The focus of this study is on the Engineering process areas, which according to Chrissis *et al.* (2011), applies to the software development industry or to any product or even service. Focusing on the software development team, the six Engineering process areas in CMMI-DEV include Product Integration, Requirements Development, Requirements Management, Technical Solution, Validation and Verification.

CMMI-DEV constellation is the only maturity model that addresses within the scope of the study. The purpose of CMMI-DEV is to assist software development teams and organisations to improve their software development processes. CMMI-DEV is for organisations that develop or maintain their own software or outsource it to service providers. Jones (2010) advices software clients who prefers software outsourcing to require some kind of proof of capability, such that the vendor be at or higher than level of a CMMI of the SEI.

The stage representation of CMMI is as depicted in **Figure 2.8**.

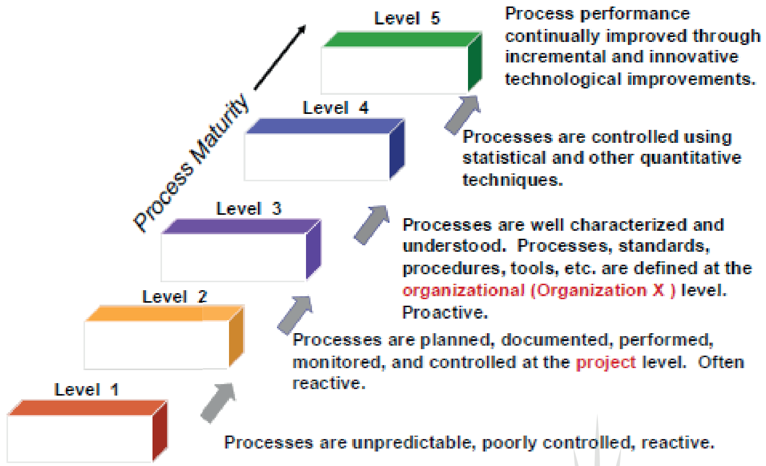


Figure 2. 8: *Maturity Levels for CMMI staged Representation (Kaur, 2014)*

According to Chemuturi (2011), SEI is the organisation that grants a capability maturity rating; level 2 is the minimum rating awarded and level 5 the maximum. In CMMI adoption, an organisation starts out at level 2 and graduates to level 5 over a period of three years (Chemuturi, 2011).

CMMI is the regulated model for evaluating and improving the development processes for products such as software and systems (Ehsan *et al.*, 2010). The current version of CMMI is 1.3, and CMMI is a successor of CMM.

CMMI-DEV addresses product development as a whole from analysis of requirements, design of product systems, management and coordination of the product systems and their integration (Galin, 2004). Other factors that affect the development of software products are defects or bugs which can be detected in requirements, design, code and testing phases. When it comes to software defect, CMMI higher levels remove more defects than lower levels. **Figure 2.9** represent the status of defect removal by each CMMI level (Jones, 2015; Tsui, Karam & Bernal, 2014).

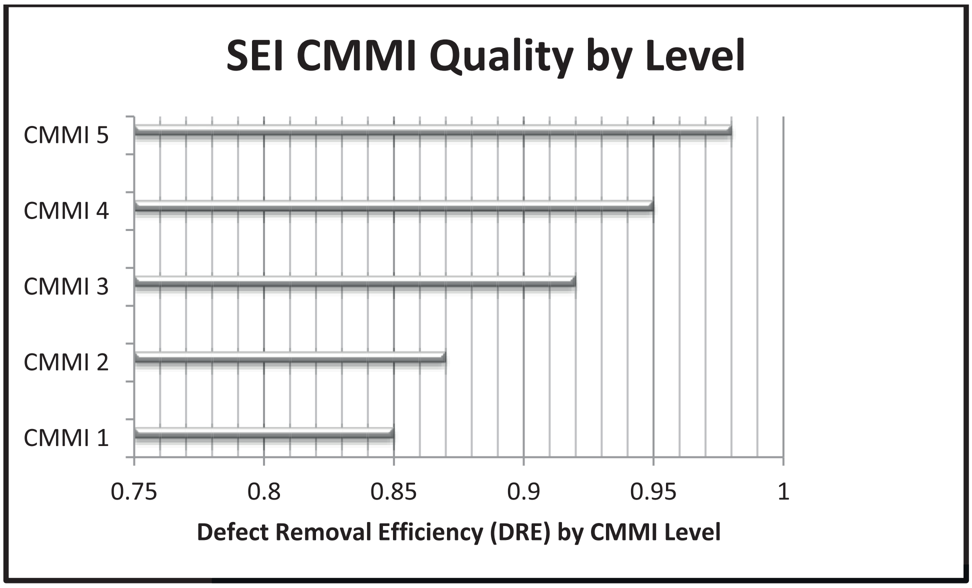


Figure 2. 9: *CMMI and Defect Removal Efficiency (DRE) (Jones, 2015)*

As shown in **Figure 2.9**, the higher the CMMI level, the higher the defect removal efficiency. Since the CMMI approach is highly recommended and sponsored by Department of Defence (military) and US Air Force, it is very important to attain CMMI level 3 or above 3, otherwise a software development company cannot be awarded a contract to serve the military or defence. In Southern Africa, there is no rule for any government department or private sector that stipulates a need for specific CMMI level as a prerequisite for tendering for a software project.

To move from level 2 to higher levels, requires the organisation to take actions to make sure that if it moves up, the organisation must have the right people to undertake a project. McConnell (1993) has reviewed projects accomplished by organisations at various levels for a typical 500,000-line code of a software project. The benefits of higher maturity levels for a typical 500,000 line code project are shown in **Table 2.7**.

Table 2. 7: *Effect Of CMM Level on Software Development of Code (Level) (McConnell, 1993).*

SEI Level	Development Cost ($million)	Development Time (months)	Product Quality (Defects / KLOC)	Productivity (LOC/Hour)	Productivity ($/LOC)
1	33	40	9	1	66
2	15	32	3	3	30
3	7	25	1	5	14
4*	3	19	0.3	8	6
5*	1	16	0.1	12	2

It is clear from **Table 2.7**, which shows the benefits and results of obtaining higher maturity levels, that adopting capability maturity principles pays (Olson, 2004). Keuten & MacFadyen (2007) have pronounced that their quality objectives were met and the numbers of defects were reduced by using CMMI. Humphrey (1997:2) made these comments about software quality: "*Software suppliers do not generally take responsibility for the defect content of their products. They often even ship products that contain known defects, and they commonly charge customers for a significant part of the costs of fixing these defective products. The public is increasingly aware of and unhappy with these practices. Software is routinely blamed for common problems in almost any industry that serves the public, and the public has come to expect software to perform badly*". This shows the importance of evaluating the maturity states of the development team or software supplier. Customers must be concerned with the maturity level of their suppliers for the sake of quality products. Using the sophisticated development approaches such as TSP to guide software and systems development projects has turned out to be highly effective (Chrissis *et al.*, 2011:8). The TSP projects are normally delivered on schedule, within budget and with extensively improved quality and efficiency (Chrissis *et al.*, 2011:8; Jones, 2010:324).

TSP approach satisfies many of the criteria for CMMI level 5, which is the highest CMMI level (Jones, 2010). According to Jones (2010), a small software project has less than 1000 function points, and a medium between 1000 and 10 000 function points. Large projects have more than 10 000 function points.

2.6.2. ISO/IEC 15504

ISO has a collection of larger standards that covers the range of domains as follows:

- ISO 9126 is for the evaluation of software quality;
- ISO 20926 is a functional size measurement method; and
- ISO 26513 is followed by testers and reviewers for documentation prepared for the user.

According to Coletta (2007), the standard ISO/IEC 15504 was developed for just performing assessment on software and systems processes. ISO/IEC 15504 is also known Software Process Improvement and Capability dEtermination (SPICE). ISO/IEC 15504 is the second largest adopted maturity model after CMMI, which is relevant to software development (Von Wangenheim *et al.*, 2010).

ISO/IEC 15504 is a software and systems engineering areas oriented model and it assesses the capability based on Software Lifecycle processes and Systems Lifecycle Processes (Coletta, 2007).

The major components of ISO/IEC 15504 are:

- 3 process categories
- 9 groups
- 48 processes
- 6 capability levels

These levels, which were defined by Hwang (2009), are shown in **Table 2.8**.

Table 2. 8: *ISO/IEC 15504 Capability Level (Hwang, 2009)*

Capability Level	ISO/IEC 15504 Capability Level Description
Level 0 Incomplete	In this first level, organisation may failure to achieve the main purpose of the process. The work cannot be associated with the end-products in advanced, the processes followed cannot guarantee a successful product.
Level 1 Performed	The success has no guarantee at this level, but the process will deliver its mandate. Repetition of previous successful projects does not guarantee the same achievement as this level.
Level 2 Managed	Level 2 is standardized, specific standards and requirements are followed accordingly in order to deliver work planned.
Level 3 Established	Now the processes are defined and documented. At level 3 the process is using defined process as a good software practice. The process is implemented using the approved versions controls.
Level 4 Predictable	The organisation rely on its defined processes for high project performance.

Level 5 Optimizing	Now processes are optimised to meet business goals. The processes can be used to reassure current and future performance. Also continuous improvement is highly expected.

The final product of the SPICE is ISO/IEC 15504, which assesses and improves software processes; the standard also integrates processes of ISO 9000 and CMMI-Dev (Grottke, 2002:10). **Table 2.8** is pictorially presented in **Figure 2.10**.

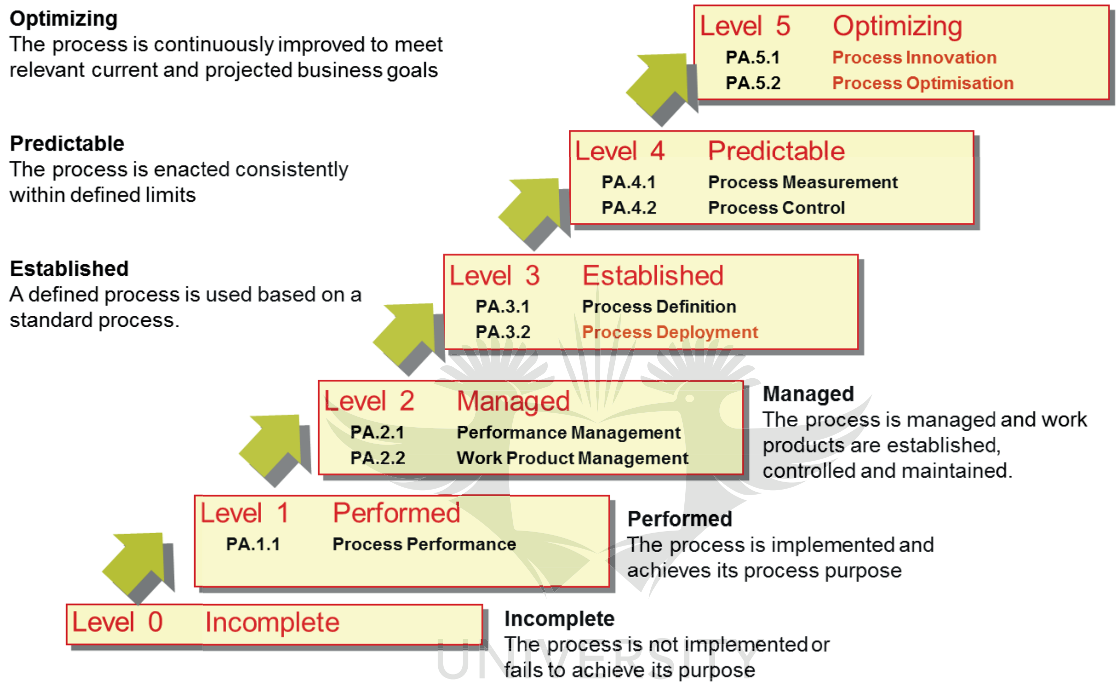


Figure 2. 10: *ISO/IEC 15504 Maturity Model (Hwang, 2009)*

The capability levels defined by ISO/IEC 15504 framework are developed to be applied openly to all types of processes (Coletta, 2007). When compared with CMMI, the processes of a maturity level of ISO 15504-7 covers several process areas that are part of other maturity levels of CMMI (Pino *et al.*, 2010). The latest version of ISO/IEC 15504 addresses issues that are connected to the assessment of organisational maturity instead of only software development maturity.

2.6.3. AGILE MATURITY MODEL

Just like the majority of maturity models, the design and development of the Agile Maturity Model (AMM) was based on CMMI. The maturity of AMM is therefore highly aligned with that of CMMI. AMM is a software maturity model for agile software development environments. After using Agile Methods that have successfully been used in other industries, the need for new a maturity for organisations in the software industry was realised and discovered. The model promises no overtime and customer satisfaction.

Figure 2.11 shows how AMM matures from an initial stage to a sustained level; the activities of each level are also listed.

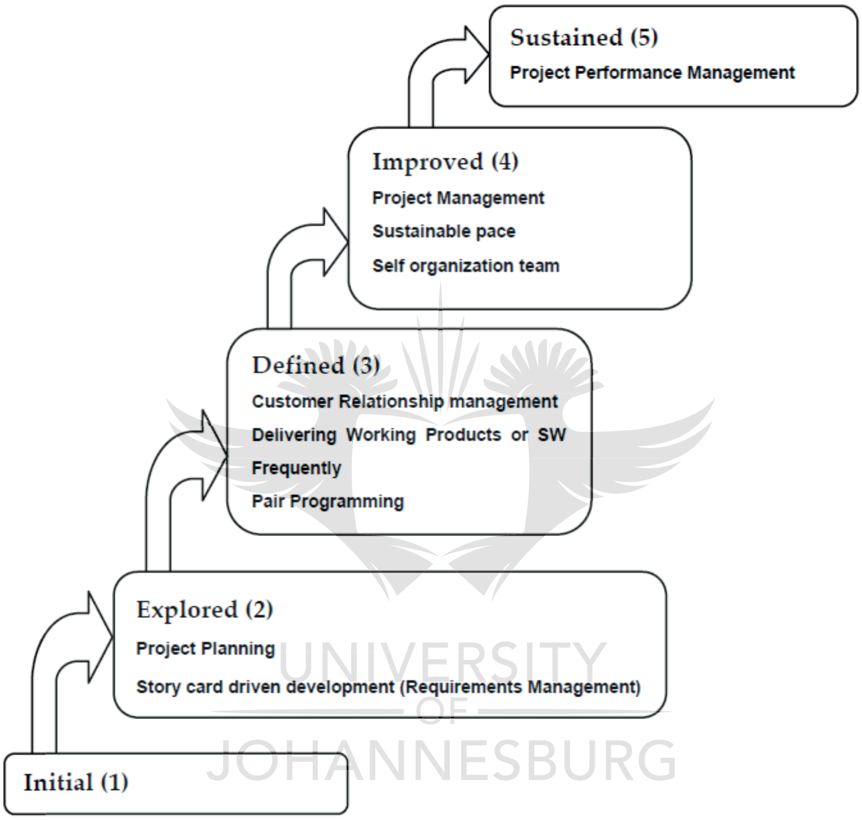


Figure 2. 11: *Agile Maturity Model (AMM)(Patel and Ramachandran, 2009)*

These Agile maturity stages are explained in Table 2.9:

Table 2. 9: *Agile Maturity Model level description (Patel & Ramachandran, 2009)*

AMM Levels	Level Description
Level 1 Initial	Initial level is an unstructured level. Process improvement goals are not accommodated. The software development team works abnormal hours,

	there is no stable environment for development. Success at this level depends on the people assigned to the project.
Level 2 Explored	Level 2 is more structured, and helps the programmers and customers to identify common problems that are related to planning, requirements engineering and onsite customer by both learning from experience and historical projects success and failure.
Level 3 Defined	At level 3, maturity continues to help developers to identify and improve problems reported by customers using customer relationship management (CRM). Improves programming, testing software and improves communication. Companies also promote pair programming and delivery of working product at this level.
Level 4 Improved	AMM level 4 promotes project management, stick to normal working hours and empower team. AMM level 4 development team must be able to organise itself, and focus on continuous improvement.
Level 5 Sustained	At AMM level 5 company continues to improve project performance and defect preventions. The level also addresses issues of customer and developer's satisfaction.

Table 2.10 illustrates the maturity models of CMMI representations, SPICE (ISO/IEC 15504) and Agile Maturity Model (AMM).

Table 2. 10: *Different Representations of Software Development Maturity Models (Fontana et al., 2014; Kaur, 2014; Mutafelija and Stromberg, 2008:29; Patel and Ramachandran., 2009 and Tsui, et al., 2014)*

	CMMI Continuous (Capability levels)	***CMMI Staged (Maturity Levels)***	***ISO/IEC 15504 Continuous Model***	***AMM Staged***
Level 5	Optimizing	Optimizing	Optimizing	Sustained
Level 4	Quantitatively managed	Quantitatively managed	Predictable	Improved
Level 3	Defined	Defined	Established	Defined
Level 2	Managed	Managed	Managed	Explored
Level 1	Performed	Initial	Performed	Initial
Level 0	Incomplete	- - - - - - - - - -	Incomplete	

Both CMMI–Continuous and SPICE implement the same number of levels with different names on level 4 and 5. AMM levels ranges from 1 to 5, the model follows CMMI staged representation. Both Staged CMMI and SPICE ISO/IEC 15504 assess the maturity of the organisation as a whole, and continuous CMMI is process areas oriented. ISO/IEC 15504 has both continuous and staged model. Models of each representation have its own advantages. The staged representation model is suited for marketing purposes as it provides a single process maturity rating (Peldzius & Ragaisis, 2011). The advantage of continuous representation model provides enough detailed assessment on how well the organisation's processes are performed (Peldzius & Ragaisis, 2011). According to Sukhoo *et al.* (2007): "*Most maturity models, for example Capability Maturity Model Integration (CMMI) and Kerzner's maturity model, have five maturity levels*". In order for an organization to move from lower maturity level to higher maturity level, it takes some years and couple of steps to achieve that. Software development organizations can obtain a maturity rating from other maturity models such as the Testing Maturity Model, SPICE, Kerzner Model, People Capability Maturity Model, IT Service Capability Maturity Model and others (Chemuturi, 2011).

Organisations from different industrial sectors such as banking, software, defence, aerospace, automobile manufacturing, and telecommunications, use CMMI for Development (Chrissis *et al.*, 2011:18; CMMI-DEV 1.3, 2010). CMMI is recognised as the best practice that gave birth to other software maturity models. The CMMI process areas that specify software development projects directly is configuration management. CMMI and ISO 9000 address similar project management issues, and when it comes to quality and process management they have common interest (Kaur, 2014). The CMMI was developed to support the software industry specifically. Hwang (2009) concluded that CMMI and ISO/IEC 15504 provide good a strategy to assess an organisation's software development capability, but they are too heavy to be applied on small-medium enterprise organisations.

For the software development industry of the world, CMMI is required more often in the United States; other models are recommended in many countries such as Australia, Japan, China and India (Chemuturi, 2011). CMMI is a process improvement model. ISO/IEC 15504 is of international standard and CMMI has become a *de facto* standard (Peldzius &Ragaisis, 2011; Garzás *et al.*, 2013). The software development industry mostly uses CMMI (Marnewick, 2013). Software development maturity models are considered the backbone of any software projects (Kaur & Sengupta, 2013). According to Von Wangenheim *et al.* (2010), most of available maturity models are assembled and designed from the CMMI framework and the standard ISO/EIC 15504 (SPICE). In other countries such as Lithuania, the private sector supports CMM/CMMI and the government promotes and supports ISO/IEC 15504 models (Peldzius & Ragaisis, 2011). The accessibility of maturity models is different; ISO/IEC 15504 comes with a cost while CMMI is freely available. Research shows that smaller companies often times experience difficulties relating to ISO/IEC standards to their business needs and practices (Larrucea *et al.*, 2016). At the end of the day, each and every software firm or organisation must produce a high quality software. According to Marnewick (2013), the current South African local maturity level does not differ from international maturity levels, which operate at the average of level three.

2.7 RELATIONSHIP BETWEEN PMM AND SDM

There are many existing approaches that can be followed to assess the maturity level of organisations in project management (Klaus-Rosińska & Kuchta, 2017). Maturity models are developed with a common purpose, which is to improve the maturity of the organisations that use them, and to improve process areas. The current section explores the relationship between project management and software development maturity. Software development maturity model and Project management maturity models have similar goals but follow different methods. The main focus of software development maturity models is process-oriented; CMMI model is an example of process-oriented model. Some models such as OPM3 and P3M3 are organisational-oriented maturity (Spalek, 2013).

Literature on organisations that applies project management and software development maturity model is still developing. Also, the literature of merging of project management and software development maturity model for the same software firm or organisation it is still emerging. Shelton (2008) does not support the implementation of Agile and PMI model at the organisational level, but she supports the adoption of Agile and CMMI which were created to address the same concerns of software development.

Currently available literature presents quality standards and methods such as ISO 9001, Scrum, Lean, when they are combined with CMMI and other maturity models. CMMI can be combined with many methods; when combined with Scrum successful performance was realised (Jakobsen and Johnson, 2008). Sutherland, Jakobsen & Johnson (2008) have demonstrated that the combination of Scrum and CMMI is more powerful than each of them separately. Other organisations have combined multiple maturity models (CMMI, OPM3), like the situation of Siemens Industry-Industry automation whereby combined maturity models have been applied leading successful performance. When are applied, both OPM3 and CMMI have demonstrated some success in improving organisational results (Keuten and MacFadyen, 2007).

According to Farrokh & Mansur (2013), maturity models are used in project management and other disciplines such as:

- Software development (Paulk *et al.*, 2003)
- Product development (SEI)
- People capability maturity
- Business development maturity model

Since this study is focussed only on software development and project management, other maturity models will therefore not be discussed. In this section, a combination of CMMI and OPM3 will be presented since OPM3 is regarded as the best maturity model in project management maturity and CMMI as the leader in the software development industry. According to Keuten & MacFadyen (2007) CMMI and OPM3 addresses sustainable performance improvements, which is required by any organisation with a desire to remain competitive. CMMI was selected based on a remark Eickelman (2003), which mentions that many organisations in the software industry have adopted the CMM since its inception. OPM3 was selected based on Farrokh & Mansur's (2013) stance, which touts and recommends OPM3 as the promising maturity model for organisations that focuses on project, portfolios or programs and which provide a competitive advantage for the organisation based on its approach. CMMI is strictly focussed on software (Marnewick, 2013) while OPM3 has a different scope; OPM3 is focused on the whole organisations at large. CMMI addresses software engineering aspects while OPM3 addresses all types of projects but does not specify best practices for software industry environment. According to Keuten & MacFadyen (2007), organisations that are using OPM3 would most likely benefit from using CMMI if they perform technical projects that require software or system engineering.

Since CMMI-DEV and OPM3 are focused on software and Project, Program and Portfolio, respectively, the opinion of the researcher, which is the main argument presented in this section, is that the organisation can apply both project management and software development maturity. Other than reaping more benefits and achieving a higher success rate, that particular organisation stands to gradually improve on its maturity level. OPM3 has low cost and can be applied to any type of industry sector. CMMI costs are high and are typically applied to the software industry only. CMMI covers project management and program management but does not cover portfolio management. Therefore, when a company applies both models, program, portfolio and project management will be covered. If activities covered by project management maturity are combined with activities of software project maturity, then a strong better maturity model can be formed if both maturity models cannot be implemented.

Table 2.11 shows the process categories of each model; CMMI and OPM3.

Table 2. 11: *OPM3 and CMMI process categories/domains (Nazar & Abbasi, 2008)*

CMMI process categories	**OPM3 process domains**
Project Management	Project Management
Process Management	Program Management
Engineering	Portfolio Management
Support	

Both maturity models address different processes, the only process is addressed by both maturity models is project management. CMMI does not recognise portfolios, and OPM3 is not addressing engineering processes. This suggests that if an organisation introduces CMMI to support a currently existing model of OPM3 or adds a CMMI oriented OPM3 to their organisation, that organisation has a good chance of achieving higher success rate in software development projects. PMI's OPM3 model can be used within CMMI. (Nazar & Abbasi (2008) have found both maturity models compatible; they are well-suited to be implemented on the same organisation. Whereas users of OPM3 have reported that they have achieved higher levels of customer satisfaction, users of CMMI have on the other hand reported that they have achieved reduction in software costs either directly by streamlining processes or indirectly by performing less rework (Keuten and MacFadyen, 2007).

Most software practitioners are familiar with 5 level maturity models, but some of the software development maturity models are represented by 3, 4 or 6 levels; the same applies to project management maturity models. Other software maturity models can be applied to a specific industry where the product is not a software (e.g. mobile devices), although most processes are software development oriented (Coletta, 2007)

Managing a software project is a challenge for practitioners as compared to managing projects from other industries. The CMMI model was designed for other industries, not for software development industry; the CMMI model was also developed to compare other processes of organisation with the best existing practices designed by other organizations (De Souza and Gomes, 2015). CMMI was developed for many other things such as to assist software and service organisations. CMMI was chosen based on its popularity since a number of global software firms have adopted this maturity model (Subramanian, Jiang and Klein, 2007). Software development for bigger software products, especially those products belonging to government or enterprises, require highly structured project management maturity model (Ahmed, 2012).

Kelsey (2006) stated that in high-maturity organisations, the software project as a whole is measured as if it were a complete business process in itself: it is effective (defines on time), it is efficient (meets budget), and it is profitable (results in margin or profit).

The maturity models are useful for determining which projects can be done in-house and which projects require assistance from contractors (Grobler and Steyn, 2006). The customers need assurances that the software provider has some expertise and high standard to provide the required software. The client or agent standing on behalf of the customer has to ask few questions concerning supplier's maturity status; these questions may include:

1. How do they handle change requests?
2. Do they use a defined development methodology such as Agile, Waterfall, RUP or any recommended? or
3. Do they follow SMART, Kerzner Model, CMMI, PRINCE II or no Project Management Maturity model at all?

The response to the above-mentioned questions are key to impressing the customer. Software customers should avoid allocating their software projects to suppliers who are not certified by international standards and local authorities. Customers should assess potential suppliers for their credibility as software providers. The customer contract may specify that "the software development must be CMMI level 2 or 3 and above" (Chrissis *et al.*, 2011:115). If the supplier states that they are rated CMMI 1 or 3, then the customer knows what is expected from the end product.

Organisations can achieve sustainable performance improvements by embracing either CMMI or OPM3 or both (Keuten and MacFadyen, 2007; Mani, Lyons and Sriram, 2010). Using both models together may prove to be very beneficial to the organisation (Keuten and MacFadyen, 2007). OPM3 and CMMI models are regularly updated, therefore, in order to increase performance which is not available when assessing organisation using one, use both to assess the organisation. Although there are no available figures to base the argument, the researcher's argument is based on available literature from Jakobsen & Johnson (2008), Keuten & MacFadyen (2007, Mani, Lyons & Sriram (2010), and Nazar & Abbasi (2008). According to Sutherland, Jakobsen & Johnson (2008) there is a relationship between software development maturity and project management maturity since they complement each other.

2.8 CONCLUSION

Although it is very rare to predict in advance whether the project under development will either succeed or fail, there are usually small signs that show the possible outcome of the project. Maturity models were not originally designed to help organisations deliver projects successfully or make projects consistently predictable, but to assist organisations to be process oriented when they perform projects routinely. The idea of project maturity is partially linked with the potential of a project to succeed.

A maturity model such as CMMI was intentionally developed to provide a disciplined framework, which can solve both software management and engineering process issues (Chrissis & Weber, 1993:51). It is not necessary for customers to encounter requirements, design, coding, or even in project management related problems, all they need is an error free software project. There are many software industry clients across the worlds that are looking for quality software products; others are concerned about the software development maturity status of the development team. Schulmeyer (2008) has revealed that by the time software product leaves the door of the supplier/provider, there is nothing much the development team can do to harness quality. Such situation is different to projects of other industries.

According to Andersen & Jessen (2003), we will never find an organisation that is matured enough; this is true because software development challenges the developers. Software project industry is a very unique industry that requires different skills to manage compared to other projects such as banking or construction (Phillips, 2004, pp. 2–10). During the development of a product such as software or a piece of software, the software development team focuses on coding the software, and this requires extensive skills. Therefore, project management maturity is there to assist an organisation to choose appropriate development methodology.

Irrespective of some differences in terms of size (small, medium or large) and complexity, the project fundamentals are the same for all types of projects. Like other industries, software projects have similar project characteristics. Software project is also a team activity. Although it can, for example, take world health practitioners a few months to contain an Ebola outbreak, it can take decades for IT/Software practitioners to find a solution for project failure or challenges. Developers must be familiar with more than one methodology and Organisations must adapt more than one maturity model, because the world is changing very fast.

Projects managers can now calculate the risks of a project at the beginning of the project. The software industry is occupied by small and medium software companies and large organisations, but all of them are expected to produce high quality products for their potential customers and the market. Any software development company should embrace CMMI, because it allows other models to be implemented within it. Organisations have options to choose their preferred maturity models, but models such as CMMI are well structured. CMMI and OPM3 were not developed to be competitors, they can be used together and the any organisation that is familiar with the one model should be able to easily adopt the other model (Keuten and MacFadyen, 2007). Finally, Andersen & Jessen (2007) states that in software development it is always impossible to reach the final stage of development.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 INTRODUCTION

In this chapter, the research methodology and design employed in this research study is discussed. The chapter specifies how to go about finding a solution and specifies a road map. The purpose of this research methodology chapter is to provide step-by-step guidelines on how the study will be conducted, so that the results can be justified to the reader.

The goal of research methodology chapter is to investigate and examine the concepts of research and research methods that are appropriate when conducting the research study.

Objectives formulated based on the above-mentioned goal of this chapter are broken down as follows:

1. To examine the concept of research
2. To access the process of producing a research plan
3. To explore the possible research paradigms and select the appropriate paradigm for this study.
4. To explore the various research strategies and time frames.
5. To select the appropriate sampling technique.
6. To define the data analysis mechanism employed in the study.
7. To explore and select a model technique for this study.

The research methodology chapter is made up of seven sections and the structure is as follows: Section 3.2 discusses the concepts of research and defines the research. Section 3.3 elaborates on the nature and significance of the research. In section 3.4, the research strategy is selected and reasons behind the adoption of the selected strategy are highlighted. Information presented here will be used to collect information about the maturity status and software projects outcomes. Section 3.5 proposes how to collect data and discusses the reasons why data was collected in the described manner. Section 3.6 present the sampling techniques and different types of sample sizes of the quantitative study. Finally, section 3.7 covers the framework for data analyses used in the study, and the theory is revelled in section 3.8.

3.2 THE CONCEPT OF RESEARCH

Sekaran & Bougie (2016) simply defines research as the process of finding a solution to a problem after a thorough study and analysis of the situational aspects. Research could discover new facts. Research starts with a question or problem description. Researchers at different organisations analyse issues that trouble them at workplaces and find a solution to the problems. Research is a process of enquiry that requires a series of activities to discover the truth, which is the main aim. The procedures to collect and analyse data and reach conclusion or truth are the elements of research (Khanzode, 2011). Doing research means we do not run in to conclusion.

Research is a search for new knowledge. People embark on research in order to discover things in a logical and efficient way and to increase their knowledge base (Saunders *et al.*, 2012). Melville & Goddard (1996) define research as a never-ending process of discovering new things and creating new products. The products such as blackberry and latest social media applications are the products of research. According to Kothari (2004), the purpose of research is to discover answers to questions by means of applying scientific procedures. Research requires gathering and interpretation of data. Other studies are crucial and involve higher costs, because they assist management to make important decisions (Sekaran and Bougie, 2016). Surveys and other research strategies are capable of revealing new information.

3.3 RESEARCH DESIGN

A research design is the plan according to which research participants are selected and information is collected from these participants (Welman & Kruger, 2001:46). Remenyi & Murray (2014) define research design as either the process of producing a research plan or even statement of the finished product that describe how the research is intended to be conducted.

The research design involves sequential steps that clarifies the research aim, of which are the definition of research problem, designing of research, collection and analysis of data and finally reporting the findings (Khanzode, 2011). A research design can be regarded as a guideline of collection, measurement and analysis of data, which are directed by the research questions of the study. The research journey begins by establishing the research elements proposed by Eriksson & Kovalainen (2011), which are illustrated in **Figure 3.1**.

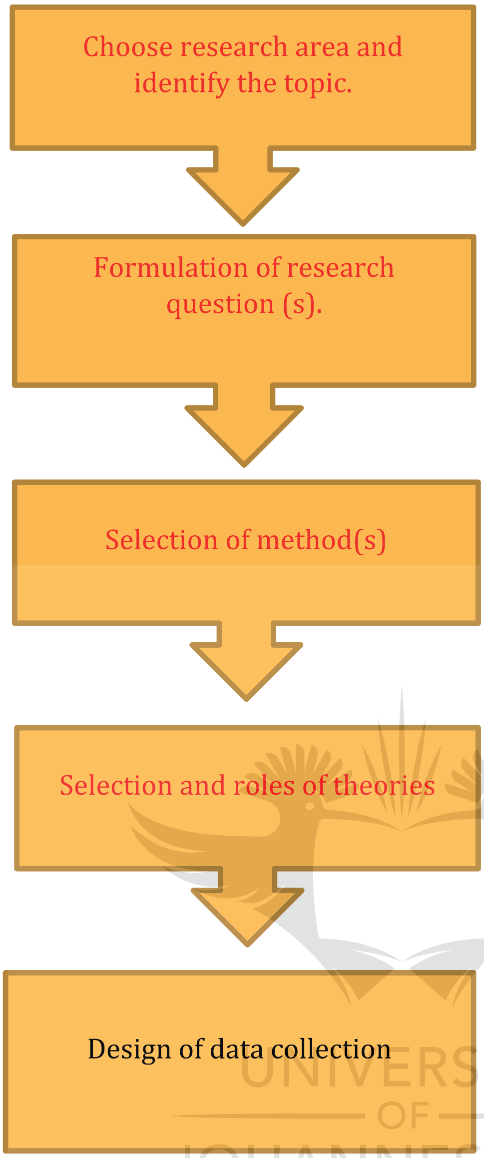


Figure 3. 1: *Elements of the research design (Adapted from Eriksson and Kovalainen, 2011:26)*

The researcher becomes aware of the problem and formulates the topic to address the problem found; one of the sources of research problem is the research that was done before. The researcher has a generic interest in the topic identified. The second element in research that comes after finding a research topic is to understand a research problem and gather current available literature related to the problem to formulate the research question (Eriksson and Kovalainen, 2011; Goddard & Melville, 2001). This must be done since the researcher might find that the same problem was resolved or addressed a long time ago by local or international research community. Upon completion of the two elements, the researcher has ideas about the possible research questions, the purpose of the research and the reason to carry out the research.

This study is descriptive since it describes the characteristics of a situation that is followed by the collection of data (Sekaran and Bougie, 2016).

The third element covers the choice of the research method(s) that is/are relevant for answering the research question formulated in the previous research step (Eriksson & Kovalainen, 2011). The appropriate paradigms employed to collect data are discussed in section 3.4.

The fourth element addresses the selection of required theories that support the argument and drives the need of research. The relevant theories were used to formulate the research questions and drive the purpose of the study. The research question directed the study to quantitative research design paradigm. The final element covers the techniques of data gathering. According to Eriksson & Kovalainen (2011), the designing of data collection must include the type of data, how to gather data, how to gain access and to plan the analysis, etc. As the positivism researcher, deductive data analysis will be adopted (Maree, 2016). The statistical terms that will be used to investigate the research questions will be defined and discussed.

3.4 RESEARCH PARADIGMS

Different philosophers and researchers view the world and science differently. Some philosophers prefer to follow alternative approaches to research such as pragmatism, positivism, constructionism and critical realism (Sekaran and Bougie, 2016). This sub-section is focussed on modes of reasoning called positivism. Science and engineering are typically based on the positivism methodology, which require a highly structured processes (Gordon, 2015). Positivists are concerned with the reliability of observations, and they use deductive reasoning to put forward theories that they test (Sekaran and Bougie, 2016). Positivists want to test cause and effect relationships through observations (Sekaran & Bougie, 2016). The key requirement for the positivism approach is that data must be reliable and replicable. Instead of using the word "*quantitative*", researchers use the word positivism; in the same breadth the word "*qualitative*" is substituted with the word "*interpretivism*" (Biggam, 2015).

The research question determines the research paradigm to use. Both paradigms apply on projects that are either numerical or not (Håkansson, 2013). The scientific research requires a researcher to decide which research paradigm to adopt if not both. The choice is also influenced by philosophical preferences and the nature of the research question (Cameron & Price, 2009); whereas some questions are important and relevant to specific perspectives, they have less meaning from other perspectives.

3.4.1 QUANTITATIVE METHODOLOGY

According to Håkansson (2013), the quantitative methodology is suitable to verify or falsify theories and hypotheses. Quantitative methodology is used to collect data that is presented using statistics, graphs, pie charts and tables. Quantitative methodology breaks down data into numbers and are useful for larger populations; this methodology is also and very specific and it uses well-defined variables (James, 2015:59). The quantitative methodology requires less labour to gather and analyse data when compared with the qualitative methodology.

3.4.2 QUALITATIVE METHODOLOGY

The alternative to quantitative methodology is a qualitative methodology, which is mostly used in specific social perspectives. Qualitative methodology is connected with the interpretive philosophy. The quantitative methodology asks open questions and allows participants to elaborate more. For this reason, the qualitative methodology is more labour intensive than the quantitative methodology.

3.4.3 DIFFERENCE BETWEEN QUALITATIVE AND QUANTITATIVE METHODOLOGY

Research makes use of three primary methods or tools for collecting data, namely: qualitative, quantitative and mixed methodology (James, 2015:59; Creswell, 2003). The choice between qualitative and quantitative methodology depends on the ability of the researcher to analyse data and the nature of the research study being conducted. The influence of the methodology selection relies on the number of aspects that direct the focus of the study. According to Biggam (2015), quantitative methodology answers the "*how*" questions and qualitative methodology answers the "*why*" questions. As far as Hollister (2014) and Erasmus & Marnewick (2012) are concerned, the quantitative methodology answers questions of "*what*", "*where*" and "*when*" and it create models and theories. On the other hand, the qualitative methodology's objective is to answer "*why*" and "*how*" that relate to human behaviour. A comparative analysis of quantitative and qualitative research approaches are presented in **Table 3.1**.

Table 3. 1: *Differences between quantitative and qualitative methodology (Cameron & Price, 2009; Greener, 2008; Naoum, 2013:43; Saunders et al., 2012:162)*

Factors	Quantitative	Qualitative
Research philosophy	Positivism	Interpretive
Research Approach	• Deductive	• Inductive
Characteristics	• Tests relationships between variables. • It is measured numerically and analysed by various statistical techniques. • Data collection standardised.	• Study meanings and the relationships between research participants. • Uses variety of data collection techniques and analytical procedures. • Collection of data not standardized
Research strategies	By default, quantitative research design is linked with experimental and survey research	Associated with couple of research strategies like action research, case study, ethnography and grounded theory.
Role	Find fact based on available evidence or records	Measuring attitude based on opinions, views or perceptions.
Relationship between theory/ concepts and research	Testing or confirm	Emerge or development oriented
Nature of data	• Numbers • Hard and reliable • Require large data sets. • In the form of numbers and units	• Words • Very rich and deep. • Small sets of data. • In the form descriptions and opinions.

While the quantitative methodology is focussed on research goals that are typically deductive, objective and general, the quantitative methodology is associated with research that is typically inductive, subjective and contextual (Morgan, 2017). Both methodologies have their own logic, process and aim. According to Leedy & Ormrod (2010), variables are identified first in the quantitative methodology. Thereafter, the data is collected specifically for those related variables.

3.4.4 MIXED METHODOLOGY

When the research employs both the quantitative and qualitative research approach, such a methodology is called a mixed methodology. In a nutshell, the mixed methodology is the combination of quantitative and qualitative methodologies. Although other studies require a combination of methodologies, Woodwell (2014) has established that positivist reject the simultaneous usage of a combination of the two methods. Saunders et al. (2012) has however professed that the mixed methodology establishes the credibility of the study and produces more complete knowledge than a single methodology. Mixed methodology may be applied on situations were one attribute is an aspect of quantitative and other attribute is qualitative as well as when the alternative methodology yields insufficient data. According to Woodwell (2014), a research can start with a quantitative methodology followed by a qualitative methodology and *vice versa*.

Based on the nature and purpose of the study and looking at the resources available, quantitative methodology is appropriate for this research study. An added reason for the adoption of the quantitative methodology for this research study is because the study is driven by the theory testing. Quantitative methodology is relevant for a number of reasons. The assumption is that quantitative method reveals enough explainable required data and results. Quantitative methodology can answer scientific question and formulate theory or verify if the hypothesis is true or not.

3.4.5 MODES OF REASONING

Theory plays a major role in research, and the research has both starting and conclusion process. According to Bhattacherjee (2012), good scientific theory should be supported by facts, and theory development involves reasoning. Reasoning could be either inductive approach or deductive, otherwise theory is abductive approach. According to Balnaves & Caputi (2001), it is very important to understand the differences between induction, deduction and abduction in research methods. The research process starts with reasoning and there are a variety of reasoning. The researcher may use observation, assumptions or theory as the starting point. According to (Cameron & Price, 2009) and Woodwell (2014), the deductive research starts with existing theory and inductive research starts with the observation and derives theory from those observations. Both induction and deduction complement each other in the positivist scientific method, although induction is a preferred approach of constructionist (Cameron & Price, 2009).

When the research use both inductive reasoning and deductive reasoning, it result in the researcher rely in abductive reasoning. Therefore, the mode of reasoning of the researcher in this research is deductive, which is more common in theoretical studies Woodwell (2014). The deductive reasoning uses existing knowledge established theories to build new theory and draw conclusions about problem based on theory and logical reasons (Bhattacherjee, 2012).

3.5 RESEARCH STRATEGY AND TIME-FRAME

At the beginning of a research study, the researcher must state the time-frame and the plan of how the research question will be answered. The best way to conduct research is to establish the research strategy that will be used to tackle the research problem (Hofstee, 2006). There are a number of tested research strategies/ design frames available, which can be used, for example, a case study, surveys, content analysis, action research, experiment, ethnography and grounded theory (Hofstee, 2006). Oates (2005) have discussed six research strategies, which are suitable for computing related research projects. These research strategies are: survey, design and creation, experiment, case study, action research and ethnography.

Most research strategies that are commonly used in dissertations are discussed in the following section (Clinning, 2016; Joseph, 2014). The time-frame required to undertake the research is also important. When are the results of the research expected? Other projects take a few days, others take weeks and others take years. It is very important for the researcher to decide the time-period of the research at the beginning of the research study.

3.5.1 RESEARCH STRATEGY SELECTION

The research strategies are linked with quantitative, qualitative or multiple methodologies (Saunders et al., (2012). Choosing a research strategy is a challenge to many researchers (Saunders et al., 2012), since there is no particular research strategy that is superior to all the others (Rule & John, 2011). According to Walliman (2018), the selection of a research strategy depends on the nature of the problem addressed by the research statement and aim. Saunders et al. (2012) mentioned a key to research strategy selection, namely the guidance and objectives of the research. The most important aspect is to answer a research question. Research strategies are not mutually exclusive, it is possible to apply the survey strategy within experimental or ethnography or mix a number of different strategies that support different paradigms (Saunders et al., 2012). Research strategies have collection of methods that can be used to collect and analyse data. Six main types of research strategies are presented in **Table 3.2**.

Table 3. 2: *Research Strategies (Cameron & Price, 2009; Eriksson & Kovalainen, 2011; Håkansson, 2013; Gordon, 2015; Oates 2005; Rule & John, 2011; Sekaran & Bougie, 2016; Woodwell, 2014)*

Research strategy	Description	Advantages	Disadvantages
Action research	Action research stems from pragmatism philosophy, but it is also compatible with other philosophies like constructionist principles. In action research, the researcher is deeply involved and collaborates with participants and other researchers in the same field. Action research does not prefer any type of data collection method, although it allows a researcher to adopt multiple ways to gather data.	• It helps to produce practical solutions to business problems. • It gives a researcher opportunity to invent or bring changes to research. • Improve the way people solves problems.	• Action research does not prescribe methods of data gathering or techniques of analysis to the researcher. • Positivist researchers do not recommend action research.
Case study	A case study is a thorough investigation of a particular entity in order to generate knowledge. Case study tells a story in details. Case studies are used to generate theories in the form of grounded theories, which arose from itself.	• It requires a combination of multiple data-gathering techniques. • Both quantitative and qualitative data can be used to construct a case. • Hypothesis can be developed. • Researcher concentrate heavily on only handful participants (may be few as 3 or 4 to 5) • Case studies are suitable for theory testing and building.	• It is not useful when the research must be delivered within a short time-frame. • The alternative hypothesis developed in most cases lack support. • Researcher can interview few participants although interview can take more than 100 hours. • Researcher cannot change team's behaviour even if there is a need.
Ethnography	Ethnography is a research strategy that has been developed for the study or different cultures and cultural sense making. It starts with general interest to a community or group of people or practical problems. Ethnography allows researchers to immerse themselves in a culture the in a culture the research	• It focusses on learning new information about human relationships. • Ethnography results in understanding of some aspect of a culture in great depth.	• The research takes long time or after a year or more, which is a long period. • Conclusion of Ethnographers is always different.

	about so that they can understand it, depending on nature of the project.	• Ethnographer can combine formal and informal data collection techniques like interviewing or participant observation.	
Experimental research	Experimental research uses a larger sample size than case study. The strategy investigate the causes and effect of relationships. The theory needs to be developed first. Experimental strategy exclude all the factors that may influence an outcome in a particular way, then researcher make detailed observation of the outcome and note the factors that when removed or introduced causes changes.	• Experimental research gives more answers to many questions than surveys and case studies. • Experiments are the only research strategy that can show evidence of casual relationships.	• Experiments are repeated as many times as possible. • Participants might change their behaviour along the process of research • Have difficulty in controlling all the necessary variables. • It is very difficult to recruit a required representative sample of participants.
Survey	The survey design is a very popular strategy in any research types, it also allows a researcher to collect both quantitative and qualitative data on any type of research question. The survey includes: 1. data collection 2. designing the study 3. Prepare reliable and valid research instrument. 4. Administer survey 5. Managing and analysing survey data. 6. Report the results Surveys can be taken of an entire population, all students at a specific given college, that sample will be called census.	• Surveys are one-time. • The questions can be arranged into self-administered questionnaires • Other survey instruments like interviews can be used. • Surveys can be used on both probability and non-probability sampling.	• Surveys lacks the in-depth investigation of a case study approach. • The researcher cannot ask follow ups questions on anonymous questionnaire.

The strategies outlined in **Table 3.2** are relevant to scientific studies, and the list is not complete. The research strategies distinguished to raise the awareness of each and to allow a researcher to choose the appropriate strategy for the study. The researcher needs to choose a single research strategy depending on nature of the study. The survey research strategy is relevant to this project and was therefore employed in this study, which is associated with questionnaires method (Berndtsson, 2008). With the survey questionnaire method, the researcher can reach a large number of respondents that are knowledgeable about the issue under investigation and it is also quicker than other methods (Cameron & Price, 2009).

3.5.2 TIME-FRAME

The time-frame section analyses the time horizon of the study. The research allows the researcher to choose times to examine the research elements which depend on time period of the research. For the researcher to answer the research question, a study can be done in a period of days, weeks or months (Sekaran & Bougie, 2016). Furthermore, the study can be a one-time study or data can be gathered at multiple points in time.

In a cross-sectional study, information is collected on a population at a single point of time (Håkansson, 2013). The researcher decides what they want to study about and identify the study population (Kumar, 2011).

For longitudinal studies, data is collected over a long period of time (Håkansson, 2013). Longitudinal design may be used in situations where the same group is examined at different time intervals (Walliman, 2005). The longitudinal studies are relevant when the investigation changes due to time constraints. Time-period may be from months to years. The longitudinal study design is regarded as sequence of repetitive cross-sectional studies (Kumar, 2011). The main advantages and disadvantages of longitudinal and cross-sectional studies are presented in **Table 3.3**.

Table 3. 3: *Time frames and the respective advantages and disadvantages of research studies (Kumar, 2011)*

Time-Frame	Advantages	Disadvantages
Longitudinal studies	• Appropriate for measuring the extent of change.	• Design is very time consuming and expensive
Cross-sectional studies	• Extremely simple in design • Appropriate design for measuring the impact of a programme.	• Cannot measure change on the same population.

The cross-sectional studies method was adopted for this research study because it is appropriate for determining if the maturity level influences the outcome of the software development projects within a short period compared to a period of over 6 months or a year.

3.6 STUDY SAMPLE

In social science or information system research, it is not necessary to survey the entire population. A small part of the population can be investigated, and those results can be generalised to apply to the entire population, which the sample (small part) was drawn from. The sample represent the whole population and it reflects the characteristics of the whole population. Sampling is another means of collecting statistical information; the same information can be in the form of numbers and figures. The various stages outlined by Cameron & Price (2009) which research must go through to define a sample from a population are as follows:

Stage 1: Define the population

→ Stage 2: Decide on the size of your sample

→ Stage 3: Decide on the sampling method to adopt

→ Stage 4: Apply the chosen method

Following the identification of the population, the size of the sample must be decided. The sample size of quantitative research is usually bigger than that of qualitative research. However, sample size is also dependent on whether sampling is for the main research where response rates must be higher.

Data is produced from a population or small part of the population. A population of an IT industry is typically very big and it consists of different sections. If there is a need to investigate all the participants that form part of the industry then the study will be very expensive and time consuming. For this reason, only a small part of the population called a sample is investigated. The sample frame that represent our population are Senior Managers, Software Developers/Programmers, Project Managers, Software Architects, Business Analysts, Quality Assurers (Testers), Project Administrators, Data Scientists and all participants that come into contact with the software when the software is under development. The participants are involved in either the design or specifications or testing and implementation phases. The researcher selected the sample frame that can represent the target population for the study.

The second stage is concerned with the sample size of the study, which is determined by the nature of the study. According to Greener (2008), there is no right answers to sample size. The researcher should however consider statistics, non-response and variation of the population than the exact sample size. The sample size is discussed in more detail in section 3.6.2.

Thirdly, a decision should be taken on the selection and application of the sampling method to be adopted for the research study. The response rates are strongly influenced by the sample techniques employed by research (Cameron & Price, 2009). The various sampling methods that can be used for a research study are discussed in the section that follows.

3.6.1 SAMPLING TECHNIQUES

Since investigating the entire population is expensive and time-consuming, a population sample should therefore be used when information is needed urgently or within a short period of time. There are two groups of sampling techniques, namely probability and non-probability sampling techniques (Khanzode, 2011). Under each group, there are a variety of sampling techniques that the researcher can employ to select a sample. Each sampling technique has its constraints and there must be a reason to choose one sampling technique that must be used for the study. Probability samples are samples that every entity in the population has a known and non-zero probability of being included in the sample. A sampling technique in which the observations are not selected randomly are non-probability sample methods. With non-probability sampling, the probability of selecting certain individuals or objects cannot be known in advance. When probability-sampling techniques are used, every unit of the population has an equal chance of being selected. With non-probability sampling techniques, the probability cannot be perfectly determined and some of the units of the population have no chance of being selected. These two major types of sampling techniques are classified further in **Table 3.4**.

Table 3. 4: *Classification of the various sampling techniques (Sekaran & Bougie, 2016; Nazar & Abbasi, 2008)*

Sampling Technique		**Description**	**Advantages**	**Disadvantages**
Probability sampling techniques	Simple random sampling	Each and every element of the population has the same or equal chance to be selected for the sample. It can be used when the whole population has the same characteristics.	• Each element in the population has equal chance of being selected. • Very good when used for postal questionnaire.	• It may be expensive when it covers large geographical area and when the researcher prefers face-to-face interview. • Most of the time a complete list of all elements in the population can be required to draw the simple random sample, which is not available often times.
	Stratified sampling	The population can be divided into small segments called strata, then random samples are selected separately from each stratum using the simple random sampling, N_1 representing the size of the population of the first strata, then N_2 representing the size of the population of the second strata and so on.	• Sampling error is reduced • Each strata may use different sampling technique.	• A complete list of all population is required, • Population sizes for each strata need to be known and some other times is difficult to find it.
	Systematic sampling	In systematic sampling method, sampling begins by randomly selecting the first element, thereafter observations are selected at a uniform interval. Imagine the list as circular, and you choose random	• Easy to administer and saves time and labour. • A complete list of the population is not always necessary.	• In order to avoid biased results, the possibility of same cycle in the data must always be remembered.

		start on the circles so that you know the starting point. It is very similar to simple random sampling.		
	Cluster sampling	The use of cluster sampling technique is more appropriate if the population is large. If the population is small, then simple random or stratified sampling strategy can be used. The technique is based on the ability of the researcher to divide the population sample into manageable groups called clusters.	• It has a reliable conclusion that can stand for research question than other strategies. • The costs of data collection are very less. • The estimations based on the model are more reliable.	• Cluster sampling has a variety of stages where sampling error may rise. • The possibility of duplication of members in several clusters is very high. • Less costly, but does not offer much efficiency in terms of confidence in the results.
	Double sampling	With double sampling, the four other probability sampling techniques discussed can all be classified as one-stage sampling technique. By combining any of the four methods, a researcher can design a probability sampling technique that suit specific research. When double sampling is used in a research to collect information, at a later stage sub-sample can be retrieved from the main primary sample to examine the study/ matter in to more details.	• The study/matter can be examined into more details. • The identified subgroup provide more details of the area of study.	• Double sampling has a lot of word for participants, because it has a chance that they might be required to participate again to reveal additional information. Which individual participants may not be happy about responding the same questionnaire for the second time.

Non-probability sampling techniques	Convenience sampling	That is collecting information from members of the population who are conveniently available to provide the required information, such as political party.	• Researcher stops collecting data when the required number of participants is reached. • The technique is quick and less expensive.	• The researcher is not guided by participants characteristics or behaviour patterns, and some of the participants in the sample they may have not the required information.
	Quota sampling	In Quota sampling, researcher is guided by visible characteristics of the study population such as gender or race. Then whenever such person is seen, that person is asked to participate in the research. The process continues until quota is reached, which is required number of participants and forms the sample.	• Researcher access sample population very easily, and the location is convenient to the researcher.	• The sampling strategy is not regulated, and their details of the participants are questionable. • The findings cannot be trusted to represent the whole population.
	Purposive/Judgement sampling	Purposive sampling is necessary when the researcher want to obtain information from specific targets or specific types of people who will be able to provide required information (e.g. Scientists, or dentists, Gynaecologists)	• Information is obtained from specialized participants only. • Researcher can obtain detailed information required.	• Selection criteria only consider expects of the area of study. • Generalization is questionable. • Researcher relies on experience.
		Snowball sampling allows participants to extent the invitation to relevant people that they know from past or on social	• Respondents are allowed to suggest appropriate participants	• The more sample become larger, the more difficult become to use technique. • Heavily rely on trust.

	Snowball sampling	medias. The snowball picks up the more snow rolls downhill.	who are interested but hard to reach. • It is useful when researcher knows little about sample population or participants. • Increases number of new person's willingness to participate.	• If the entire sample belongs to a particular faction or group, then the choice of individuals may influence the study out come and the study may be biased.

The sampling techniques strategy that was adopted for this research study was snowballing because snowballing allows few individuals from the entire sample frame to act as informants and identify other members. There was no alternative way to access the required sample, hence snowball was the only viable sampling technique of choice. Strategies such as quota sampling emphasize on specific sample size, and convenience sampling requires a researcher to have access to a remote software development firm or company. Simple random sampling was also found not to be suitable for this research since this sampling technique is too general. Welman & Kruger (2001) compares snowball sampling with rolling snowball because it grows in size until it is over distributed.

3.6.2 SAMPLE SIZE

A considerable number of participants was expected for this research study. According to Oates (2005), in order to generalize the research findings, the sample must be from an adequate size. Determining whether a sample size is big enough is not a simple matter in quantitative research, and a larger sample does not guarantee high levels of accuracy (Balnaves & Caputi, 2001). While Maree (2016) has reiterated that the question of sample size is not easy to answer, Kumar (2011) believes that the larger the sample size, the more accurate the findings. Since the snowball sampling technique has been adopted for this study, the researcher expects the participants to extend the survey to other potential participants.

All non-probability sampling techniques have rules governing the sample size (Saunders *et al.,* 2012). For example, Roscoe (1975) have proposed that sample sizes that are between 30 and 500 are appropriate for most research. However, Saunders *et al*. (2012) asserts that what is important when using techniques such as snowball is the logical relationship between your sample selection technique and the purpose and focus of the research, and not the sample size.

3.7 DATA ANALYSIS

In the previous section, the researcher ensured that the quantity and quality of information required and obtained meets the needs of the study. The data analysis framework used for the study, which was adopted from Clinning (2016), Sekaran & Bougie (2016:275-297) and Walliman (2005), is depicted in **Figure 3.2**.

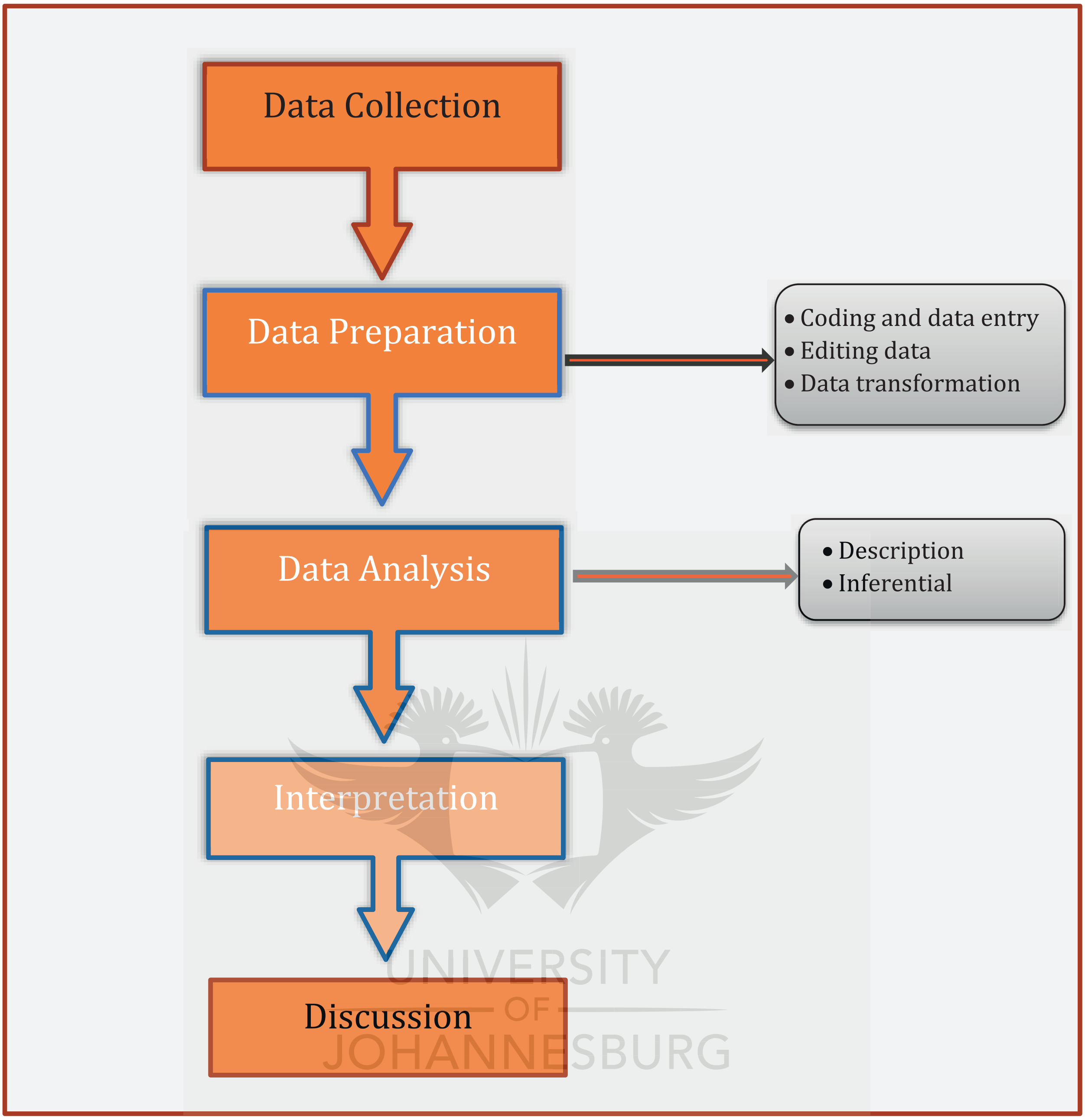


Figure 3. 2: *Data analysis framework for this research study (Adopted from Clinning, 2016:44; Sekaran & Bougie, 2013: 275-297; Walliman, 2005)*

3.7.1 PILOT TESTS

Prior to the collection of data, the instrument was pilot tested using a small sample of participants. Once the questionnaire was pilot tested, the duration to complete the instrument was determined. The pilot test includes few closed questions, which were returned by only 10 respondents. The sample of 10 respondents was regarded as sufficient and the necessary amendment was done. It was only realised after pilot testing that some organisations can have close to 80 projects within 6 months. The researcher used the pilot study to check the relevancy of question wording. After conducting the pilot study, the researcher rephrased other questions under the guidance of the supervisor and statistician. Thereafter, the questionnaire was finalised and published. A copy of the used instrument is provided in Appendix B.

3.7.2 DATA COLLECTION

Primary data was gathered directly from participants using a questionnaire. When the research instrument was finalised, a questionnaire was distributed electronically to respondents *via* email and an online link. In addition, a hard copy of the questionnaire was handed to some respondents. This means that both an online and hard copy surveys were employed in the study. The participants were selected by using the snowball sampling technique. A quantitative research methodology was used in favour of a qualitative methodology, because methods such as observation, interview and content analysis were considered not suitable for this research study.

Data was collected using a structured questionnaire that addresses the required variables to be instigated. According to Saunders et al. (2012:420), the questionnaire is either interviewer-completed or self-completed. Three data collection strategies employed for self-completed questionnaire are sharing link via email, attaching to an email a questionnaire in Microsoft (MS) Word and providing printed hard copies to those who do not prefer the above-mentioned methods (Saunders et al., 2012:420). Participants that used hard copies were found when the researcher attended the Project Management South Africa Conference; participants were given printed questionnaire by hand and the completed questionnaire were returned the following day. Data collection was undertaken during the periods 04 September and 16 November 2018. Invitation to participate in the research study included an invitation letter, an ethical clearance certificate and a questionnaire.

3.7.3 DATA PREPARATION

The crucial part of a successful data analysis is data preparation, which makes data to be ready for analysis. To process collected data successfully, technical matters related to data analysis must be covered. These technical matters, which are discussed individually in the following sub-sections, include coding and data entry, editing data, data transformation, as well as validity, reliability and ethical.

3.7.3.1 Coding And Data Entry

When raw data is collected, it must go through a coding process. A number will be assigned to each participant in order to differentiate their responses in the database. The coding approach that assigns a number like 1 to never and 5 to every time, and strongly disagree to 1 and strongly agree to 5 (i.e. Likert-scale) was used in the research study. The raw data was classified into meaningful data categories by the researcher.

Data entry was done using MS Excel and SPSS data editor, which is a tool designed to analyse data specifically (Greener, 2008).

3.7.3.2 Editing Data

After data entry and coding, data need to be edited. The editing of data is required to improve the quality of the coded data (Khanzode, 2011). The editing process takes place immediately after closing the survey. At this stage, duplicate responses are removed.

3.7.3.3 Data Transformation

Data transformation refers to the process of changing and assigning the original numerical quantitative value to another value (Sekaran & Bougie, 2013). The validity and test reliability of the instrument is assessed in the following sub-section.

3.7.3.4 Validity, Reliability and Ethical Considerations

The quality of a positivist research must be assessed for reliability and validity. There are many validity types which can used to test and measure the goodness of instrument. In this research study, the reliability of the instrument was checked and tested by using statistical approach functions such as correlation, regression and Cronbach alpha. In General, the meaning of validity is the ability of a research instrument to measure what is designed to measure (Kumar, 2011).

3.7.3.4.1 Content validity

Since the instrument is not readily available instrument for use, the researcher relied on expert advice from the supervisor to guide and determine if the instrument actually tests what it is supposed to test (Goddard & Melville, 2001)

3.7.3.4.2 Construct validity

Construct validity refers to the use of existing research instrument which was previously used, and it measures the things the researcher wants to determine

(Goddard & Melville, 2001:47). Currently there is no instrument that measures the relationship between software development maturity level and project success.

3.7.3.4.3 Internal validity

Internal validity refers to the determination of integrity of the research and quality of the research process. The researcher examines the right things and collects the required data from the software development team. Thereafter, the researcher will be able to justify whether or not the maturity level determines project success. The survey used in this research study supports internal validity, since it allows the researcher to draw conclusions about the relationship of two variables.

3.7.3.4.4 External validity

The sample was found to be representative of the population since all the participants are working within the software development industry. The participants are individuals who take part on the development of software, ranging from project manager to software requirement engineer. The research findings are only generalizable to software development (Sekaran, 1992).

3.7.3.4.5 Reliability

Reliability is a measure that indicates the extent to the instrument is error free; this ensures consistent measurement throughout the duration of the research and even if used by a different researcher. In a nutshell, reliability ensures that when other researchers use the instrument, it will produce the same results (Sekaran & Bougie, 2016). The researcher is allowed to re-use any existing research instrument, as long as the researcher can cite the source so that the reader or marker can find more information if required. Currently, there is no complete instrument that can be used to determine if the maturity level determines the project outcome. According to the experience of Devlin (2006), the most commonly and regularly used analysis is descriptive statistics, which covers frequency, descriptive and cross tabs functions).

The closer the Cronbach's alpha is to 1, the higher the internal consistency reliability (Sekaran & Bougie, 2016:293). Generally, reliability values of less than 0.6 are considered very poor; any reliability value that is higher than 0.75 is regarded as being very good. According to Maree (2016), reliability value of 0.8 is regarded as acceptable in most situations, but reliability values of less than 0.6 are not acceptable. In case of a low Cronbach's alpha, other items must be removed (questions or constructs). In this study, the measurement scales of validity and reliability were determined by calculating the values of the Cronbach's alpha using SPSS.

3.7.3.4.6 Ethical considerations

Ethics concerns researchers because they have to seek access to information provided by organisations and individuals. The study adhered to ethical values prescribed by the University of Johannesburg (UJ) to defend the integrity of the research, and all attempts were made by the researcher to act with integrity. UJ research ethics committee assessed the questionnaire before it was distributed to participants. According to Walliman (2018), research has value if it is carried honestly. Therefore, all attempts were made to treat the participants with respect during and after the conclusion of the research. Save for the email addresses shared by the participants, the collected data does not contain any confidential information of the participants. Anonymity was maintained at all times during and after the undertaking of this research. According to Saunders *et al.* (2012), ethical issues appear at different stages of the research. The study is concerned with ethical issues during data collection, analysing and reporting stages. The information required was not sensitive to the respondents. During the data collection process, the participants have a right to withdraw at any time or decline to participate (Saunders et al., 2012). During the data analysis and reporting stages, the researcher maintained data honesty on data presentation.

3.7.4 DATA ANALYSIS

The information obtained from respondents is often converted into figures and tables, and it must go through analyses and evaluation steps prior to being interpreted. The analysis and evaluation steps are carried out in relation to the research problem (Walliman, 2005) because analysing data that is not related to the aim of research is a waste of time. Since research is not a linear process, the study requires a statistical method that will help to transform numbers into useful information for decision-making purposes. Statistics is very useful and allows the researcher to understand and gain more knowledge about the risks associated with business decision making in the face of uncertainty (Levine, 2010). Statistics caters for all techniques that collect, analyse, evaluate and interpret data. According to Levine (2010), statistics has two methods used to collect, summarise, present and analyse data. The methods are called descriptive and inferential statistics. In this research study, SPSS and MS excel was used to analyse the data.

3.7.4.1 Descriptive Statistics

Descriptive statistics is a statistical technique that facilitate the collection, summarising, presentation and analysis of data (Levine, 2010). Descriptive statistics measures dispersion of data through the use of standard deviation, mean and the mode.

- Standard deviation: Just like variance, standard deviation is a measure of internal dispersion of a data set, and it equals the square root of variance.
- Mean: this is the measure of central tendency of items. According to Levine (2010), all the values play an equal role. The formula to calculate mean is the sum of the values divided by number of values. Extremely big values affect the mean.
- Mode: this is the value that appears the most in a data set. Unlike the mean, outliers do not affect the mode.

3.7.4.2 Inferential Statistics

Inferential statistics is a method that uses the data collected from a small group to reach conclusions about a large group (Levine, 2010). According to Walliman (2005), inferential statistics processes predictions through inference based on the analysed available data.

- *Correlation*: is a statistical element that is designed to quantify the degree of correlation between variables (Greener & Thomas, 2015:78). The function answers the question of whether or not the two variables are related across the sample (Devlin, 2006). Bhattacherjee (2012) defines statistics as a value that is estimated from data. Sekaran & Bougie (2013) posit that correlation is determined by assessing the variations in one variable as another variable also varies. A correlation in a study indicates the strength and significance of the relationship between project success and maturity level when they are measured at interval level. The Pearson correlation coefficient is a measure of the strength of the relationships between two or more variables (Maree, 2016); in this research study, it was used to measure the relationship between project outcome and maturity level. The project outcome at each maturity level can be determined by bar chart or scatter plot. The most used correlation statistic is Pearson's, which is denoted by r and varies from -1 to 1; a value of zero means there is no correlation between project success and maturity level (Woodwell, 2014). Values of –l and +1 denote a perfect linear relationship.
- *Variance:* this is the sum total of the difference between each value and its mean. Variance measures a scatter around the mean. Variance is always positive and measures a degree of dispersion. Variance was used to determine the consistency of response within the constructs.

- *Regression analysis:* This is used when one independent variable is hypothesized to influence the dependant variable (Sekaran & Bougie, 2013). This study proposes that maturity level influences the software development output. Therefore it is required the data that will display the relationship between software project outcome and maturity level should be plotted. After the plotting data, a linear straight-line equation is then implemented. The simple linear equation is represented as follows:

  $Y= b_0 + i_1x_1$

  where Y is project outcome, b_0 is an intercept as the project successes, i_1 is the slope and x_1 represents maturity outcome.

  Regression gives more precision when asking the question of whether a given variable significantly predicts a certain outcome of interest (Devlin, 2006).

The main purpose of parametric statistics is to identify relationships between variables (Walliman, 2005:305). Both descriptive and inferential statistics are appropriate for this research study and have therefore been adopted for this research study. According to Woodwell (2014), quantitative research is very much associated with regression analysis and statistics. Determination of standard deviation is relevant to the study. Also, both the correlation and regression from inferential statistics are deemed very important for this research study. To this end, it was envisaged that these parameters will be used for determining the correlation between project success and maturity level. Computer applications for plotting the relevant graphs were therefore used with the aim of facilitating the identification of the significant relationships between the variables.

3.7.5 INTERPRETATION AND DISCUSSION

The data interpretation phase is focussed on interpreting the data that will be collected using the selected research instrument. The data interpretation was facilitated through the use of figures and tables. The discussion is presented in chapter 4, whereby a comparative analysis of results and existing literature is undertaken. It is envisaged that the interpretation and discussion presented in chapter 4 will assist the researcher to compare the correlations of different studies and the software development projects outcomes as whole.

3.8 MODELS/THEORIES/HYPOTHESIS

According Oates (2005), there are many reasons why people are doing research. The main goal of quantitative research is to design a model or develop a theory and hypothesis pertaining to specific natural phenomena (Berndtsson, 2008). According to Bhattacherjee (2012), models are mostly used by decision makers and represents a problem, while theory attempts to explain a problem. The outcome of the research reveals the new way of looking at software project. **Figure 3.3** displays full layout of the research design that was adopted for this research study.

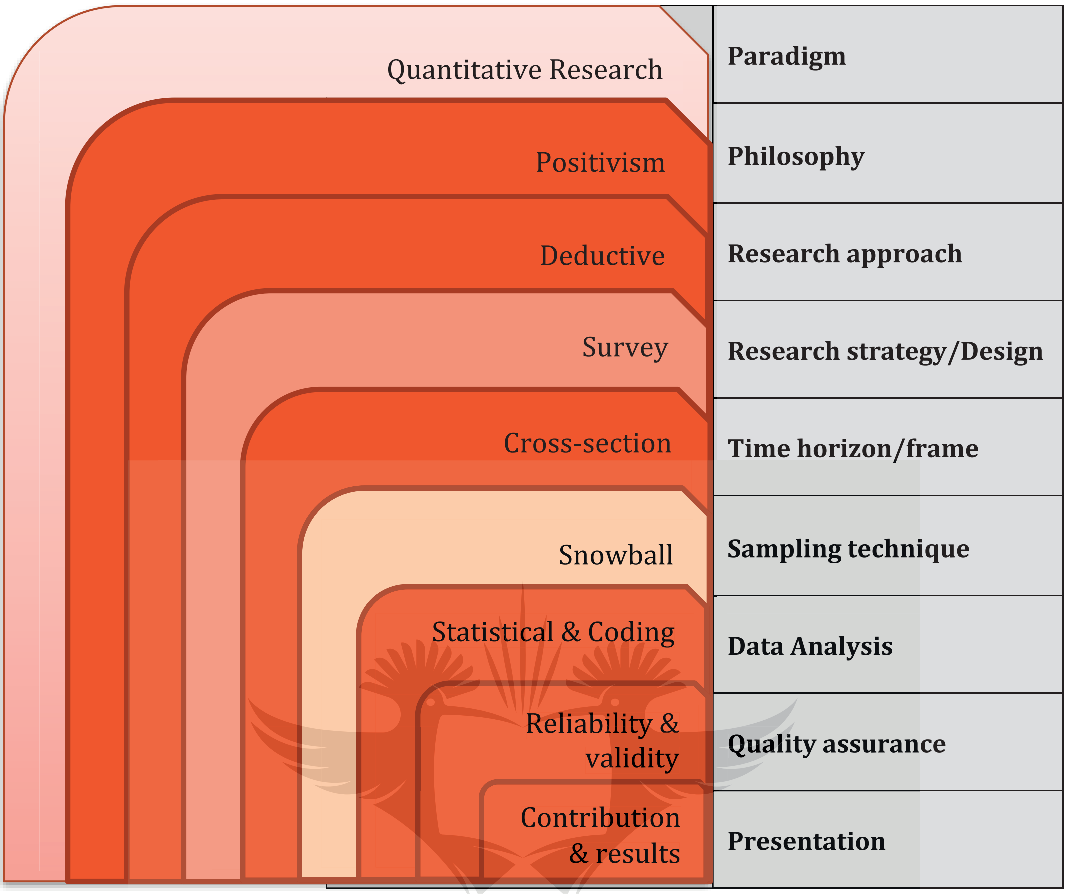


Figure 3. 3: *Research design layout*

3.9 CHAPTER CONCLUSION

This research methodology chapter has addressed the process that was followed for the selection of a method for undertaking this research study. The positivism philosophy was adopted based on the research questions of the study. The selection of deductive mode of reasoning was justified. The study used time-horizon of cross-sectional and tests theory through observation. A snowballing sampling technique was used for the study. Following collection of the data over a two-month period, the collected data was captured on SPSS. The questionnaire was constructed based on the SEI's maturity model. In the next chapter 4, the quantitative results obtained from the respondents using questionnaire are presented and analysed.

CHAPTER 4: DATA PRESENTATION, ANALYSIS AND INTERPRETATION

4.1 INTRODUCTION

This chapter follows the proposed course of action outlined in chapter three. Chapter 3 described in detail the process and purpose of the quantitative research design. Chapter 4 transforms the received raw data into more meaningful and relevant information. The type of data gathering techniques that was employed in this research study is a survey, which emphasises the use of diagrams and tables to explain and understand the research data. According to Hofstee (2006), the three main factors that constitute the body of dissertation data analysis (i.e. Chapter 4), are research findings, analysis of findings and interpretation of the collected data. The interpretation will lead to sub-conclusions, and sub-conclusions will assist the researcher to get to the conclusion of the study. Therefore, the chapter is consisted of the findings, analyses and interpretation of the collected data.

The main goal of the Chapter 4 is to explain how the research was carried out; specifically, it covers data collection and presentation as well as analysis of the results. In order to achieve the above mentioned goal of the chapter, the following objectives were compiled:

1. To explore the main reason for collecting and preparing data.
2. To examine the respondents industry background in relation to the research.
3. To analyse the success rates of software development projects.
4. To systematically evaluate the collected data against the objectives of the study.
5. To discuss how software development project success is influenced by maturity levels.
6. To draw some conclusions about the overall objectives of the study.

The structure of this chapter consists of five major sections. The first section outlines the main goal of the analysis and presentation of the results. Upon completion of the data analysis framework (i.e. second section), the third section is focussed on a data collection strategy adopted as well as the data preparation step. In the fourth section (4.4), figures and tables are used to present data that was collected using a questionnaire. An analysis, evaluation and interpretation of the collected data against the objectives of the study is also undertaken. The headings of the fourth section correspond with the questions outlined in the questionnaire. The last section is a conclusion section that is based on the analysed data. The findings are reported using the quantitative approach.

4.2 DATA ANALYSIS FRAMEWORK

As shown in **Figure 3.2**, the data analysis framework was unpacked in Chapter 3. The framework was adopted from Clinning (2016), Sekaran & Bougie (2013) and Walliman (2005). The researcher chose the methods that allowed an objective measurement of the variable of interest (Leedy & Ormrod, 2010). The phases of the framework including data collection and preparation of quantitative study are discussed in the following section.

4.3 DATA COLLECTION AND PREPARATION

A questionnaire was electronically distributed to respondents *via* email and an online link. The respondents from software development industry were selected by using the snowball sampling method. Following the testing and refinement of the questionnaire, the respondents were not expected to encounter any problems when answering the questions. The questionnaire was hosted on an online platform. Collected data was downloaded and imported into IBM SPSS statistics version 25.

4.3.1 DATA EDITING

Collected data is expected to be relevant, clean and in the correct format (Wegner, 2012). Since the questionnaire was placed on an online platform, there was no need to edit it further or even check for typographic errors. The blank responses were left blank. Editing data may result in the validity and reliability of the date being compromised. No data inconsistency, outliers and illegal codes were noted. The options such as '*I don't know*' were eliminated from the questionnaire during the design process, because the questions of Likert-scale options which have more meaning were used and coded as 1 to 5 (Strongly disagree (1), disagree (2), neutral (3), agree (4) and strongly agree (5)) to show the level of agreement.

4.3.2 DATA CODING AND ENTRY

The current section creates a clear process of coding data and capturing it for statistical application. According to Oates (2005), data is either in a numeric form or it needs to be translated into numbers before the researcher can carry out any quantitative data analysis. The first step to analyse the received data is to code it; therefore, the original character of information was transformed into numerical values. The number was assigned to the participation responses and the number was entered into a statistical application (Sekaran & Bougie, 2016). The coding scheme was designed prior to data collection, hence there was no need to code after collecting the data. The Likert-type scale of frequency of use and level of agreement were coded 1 to 5.

The collected data was downloaded onto SPSS 25. Data cleaning was performed by removing all unusable responses prior to analysis; non-responses were left blank. The following data coding system was applied, and the numbers were used to represent the responses. Likert-scale questions were coded as follows: strongly disagree (1), disagree (2), neutral (3), agree (4) and strongly agree (5). Printed questionnaire responses were captured using the SPSS Data Editor. Each row of the SPSS Data Editor represents a case and each column represent a variable (Sekaran & Bougie, 2016). The questionnaire had 212 cases and 101 variables. Hence, it was possible to compare and evaluate the data in the data set. At this point data was deemed available and prepared, and the next step was to analyse and explain the meaning of the received data. The main graphical representation that has been adopted for analysis is bar charts. This is because bar charts can show comparisons of categorical data.

4.4 DATA ANALYSIS, INTERPRETATION AND DISCUSSION

A quantitative researcher maintains the objectives of the study for purposes of data analysis (Leedy & Ormrod, 2010). The quantitative data analysed in this study is an ordinal data, the responses are categorised into Likert scale questions where the numbers are assigned to options as explained in section 4.3. According to Oates (2005), ordinal data is called ranked data, because categories are ranked. The instrument used to gather information from research elements needs to be tested for reliability and validity, hence the reason why the reliability level achieved is discussed in the next section. The design of reliable instrument for measuring capabilities of respondents require good planning (Melville & Goddard, 1996). The research is characterised using the CMMI models, and the questions were designed from the books of CMMI (Kulpa & Johnson, 2008; Kenett & Baker, 2010; Persse, 2007). The books define the goals and objectives for each process area. Overall, the maturity questions were designed based on CMMI process definitions.

The maturity questions (See sections C, D and E of the Questionnaire – Appendix B) covers software development processes, project management and the organisation as a whole. Software development process is concerned with the scope and completeness of the process and how the process is managed, measured as well as how the process can be improved. The questions are detailed in the section that focuses on software development process (see survey instrument in Appendix B). Organisation questions addresses the personnel responsibilities, organisational processes and other resources that are required for the project and other management organisations. The questionnaire is comprised of: 21 Likert-scale questions that focuses on five organisation processes; 19 questions that focuses on six project management process; and 25 questions that focuses on six software development processes. As indicated in Chapter 2, the maturity levels are defined as: Initial; Repeatable; Defined; Managed; and Optimised. The Initial level denotes the lowest level, and Optimised level is the highest level.

4.4.1 RELIABILITY AND VALIDITY

The research strategy that was appropriate for the research was adopted, and the relevant data collection techniques was employed. According to Cameron & Price (2009), data is reliable when it contributes to answering the research question, and the validity of the data needs to be determined. The responses were analysed for internal consistency reliability using SPSS. The Cronbach's Alpha was calculated and results are depicted in **Table 4.1**. These results show high reliability levels with 96 items/competencies that are divided into 21 constructs providing good evidence that the competencies used are a valid measure of project success and maturity level.

Table 4. 1: *Cronbach's alpha*

Reliability Statistics		
Scale items	**Cronbach's Alpha**	**Number of Items**
Project Outcome	0.840	5
Software Development Critical Factors	0.956	26
Software development process areas	0.963	25
Project Management Process areas	0.965	19
Organisation Process areas	0.971	21

The Cronbach's Alpha coefficient of reliability was computed for all variables measuring project outcome, software development critical factors, software development maturity processes, project management maturity processes and organisational maturity processes. The Cronbach's Alpha was easily calculated using a statistical analysis program.

The reliability of the questionnaire was checked and tested by comparing the Cronbach alpha values with those derived using STATA v15 and SPSS v25. The values derived from all the statistical functions (i.e. Cronbach's Apha, STATA v15 and SPSS v25) were similar. All the variables of the research are represented in **Table 4.1** and the items of the Cronbach alpha for each item are discussed in each section that present the item results in detail. Cronbach Alpha values greater than 0.9, which were produced for all constructs, indicate high stability and the fact that the variables used to measure project success and maturity level were indeed reliable. Therefore, each variable of the study had acceptable internal consistency reliability. Consistency shows how well the items are positively correlated to one another (Sekaran and Bougie, 2016).

Cronbach's alpha for the research study was found to be above 0.7 on all the constructs mentioned and was acceptable. The data was found to be reliable; the reliability of the data confirms that the data collection techniques used and the analytical procedures employed would produce consistent results if repeated by a different researcher (Sanders, 2012).

4.4.2 RESPONDENTS BACKGROUND

A total of 750 respondents were contacted and only 480 agreed to participate in the study. About 55% of the respondents were contacted *via* social media (i.e. LinkedIn). Upon completion of the survey, relevant data was however only collected from 212 respondents. The sample size for the study was therefore 212 in total. An overall response rate of 28% was recorded; this means that out of 750 invited respondents, only 212 completed questionnaires were received. The focus of the research was on software development project success, so the level of experience in relation was crucial and essential.

The purpose of asking respondents about their roles was to ensure that the respondents belongs to the software industry, and they have the required skills to participate in the survey so that the survey can produce meaningful results. The number of years of experience of the respondents allowed the researcher to deduce whether or not the respondents were familiar with software processes. The results are represented in a cross-tabulation form as shown in **Table 4.2**. The highest percentage of respondents was software developers (52.35%), followed by other (14.62%) and business analysts (8.49%). In the position question, there was an option for 'Other', which refers to other positions within software development that were not listed in the questionnaire. Positions that were not listed for the respondents were Programme Manager, Senior DBA and Database Administrator. Whereas 7% of the respondent were project managers, only 2.5% of respondents were quality assurers (Testers). **Table 4.2** also shows the work experience period of the target population.

Table 4. 2: *Cross-tabulation of Position/Job Title and Work experience*

Position/ Job Title	Length of Time/Work experiences							Total
	Less than 1 year	1 – 5 years	5 - 10 years	10 - 15 years	15 - 20 years	More than 20 years	Prefer Not to say	
Senior Manager	2	0	5	1	1	2	0	11
Software Developer/Programmer	10	41	28	18	6	8	0	111
Project Manager	0	3	4	2	2	2	2	15
Software Architect	0	0	1	1	1	1	0	4
Business analyst	3	1	5	7	1	1	0	18
Quality Assurer (Testers)	0	4	0	1	0	0	0	5
Project Administrator	0	1	2	0	0	0	1	4
Data Scientist	1	5	3	4	0	0	0	13
Other	2	8	5	11	3	2	0	31
Total	**18**	**63**	**53**	**45**	**14**	**16**	**3**	**212**

As already mentioned, the total sum of the respondents was reduced to 212. Three respondents did not indicate their work experience, although it was very important to indicate their work experiences in the software development industry. **Figure 4.1** shows how the breakdown of the various positions of the respondents.

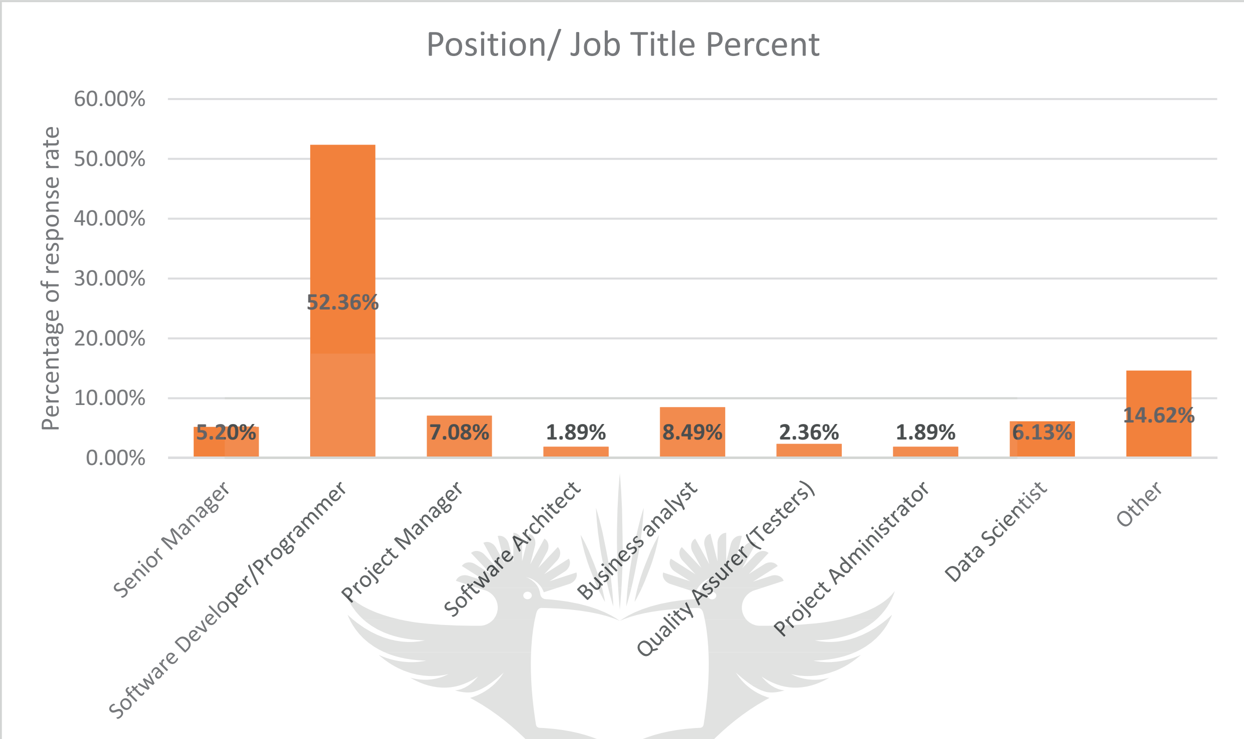


Figure 4. 1: *Position/Job title breakdown of the respondents*

The second question that characterises the respondents was work experience, which indicate the duration of the respondents on the same field. An analysis of work experience data of the respondents showed that most software developers have less than 5 years of experience. About 7.5% (n=16) of the respondents are well experienced (> 20 years experience), and 6.6% have work experience of between 15-20 years.

Since this research is focused on the software development processes, it was important to determine experience in a software development team for each respondent. The respondents work experience discovered by the study was satisfactory. **Figure 4.2** was computed to provide the overall work experience related to the software industry.

Results presented in **Figure 4.2** shows that the 29.7% of respondents have 1-5 years of experience, and 25% have 5-10 years. The third highest ranking experience category was 10-15 years. These results show that the respondents have an overall work experience of between 10 and 15 years in the software development industry.

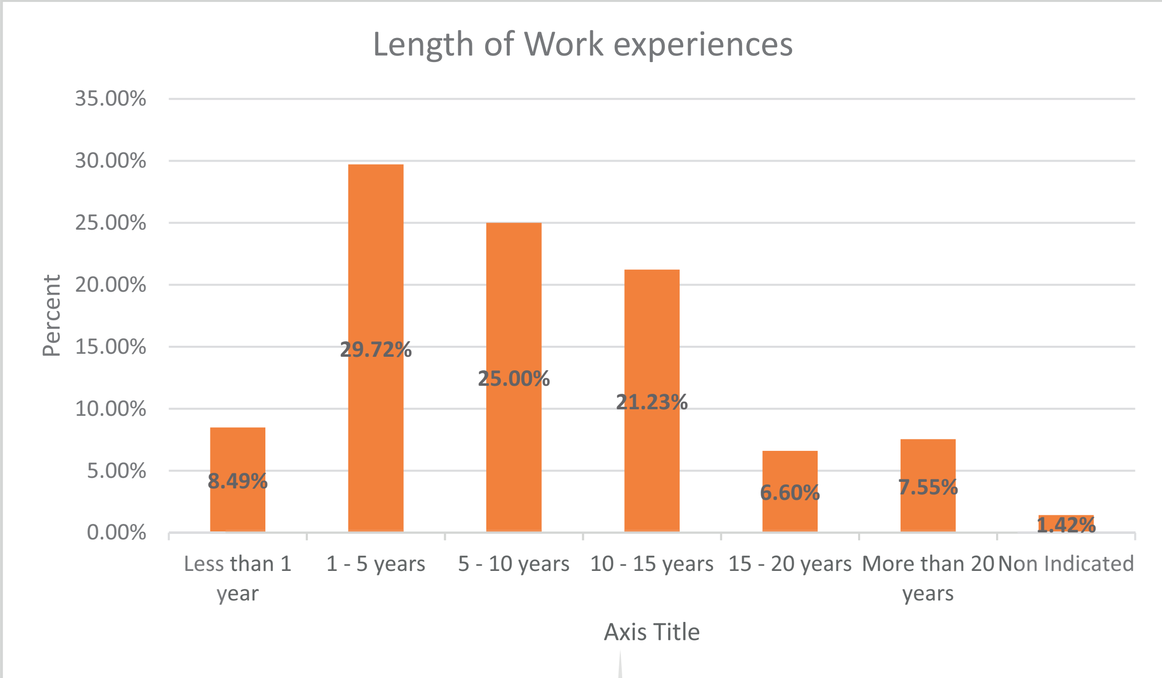


Figure 4. 2: *Work experience of the respondents*

The data analysed represents the small margin of less than 1-year experience of respondents, of which does not have influence on the overall study, including 1.4% of the respondents who didn't indicate their level of experiences.

4.4.3 THE PROJECT PERFOMANCE AND FACTORS

IT or software projects are unique in many ways. Therefore, the project performance is measured differently and the factors that affect the success of the projects are unique to each project. The performance results generated in this study were compared with the results of other longitudinal studies in academic environment.

The respondents were asked to indicate the extent/level in which they agree or disagree that each success factor is critical to the success of their software development projects. The factors were scored on a 5-point Likert scale ranging from 1 (strongly disagree) to 5 (strongly agree). The Cronbach alpha coefficient for this subset is 0.956, and is hence valid for this analysis. The respondents were presented with the software development factors and asked to rate their extent of agreement with the factor level of criticality for the success of the software development project.

4.4.3.1 The project outcome

The first objectives of the study was to determine the current software project success rate. According to *VERSIONONE.COM* (2018), the top measures of project success are based on time delivery of product, product quality and customer or user satisfaction. The study has adopted the measures indicated by *VERSIONONE.COM (* 2018)/ VersionOne Inc. (2016) and complemented success dimensions with budget as one of the success pillars. Therefore, the interpretation of project success on this study was based on three main measures of success, namely: budget, quality and time taken to deliver the software project (see **Table 4.3**). Furthermore, quality outcome was divided into three measures, which are product specifications, customer satisfaction and the use of the delivered software by the customer. Time was used to measure the duration to develop and deliver a software developed and has a correlation with a number of measures such software size. Therefore, in order to measure the success rate of the software development projects, the respondents were asked to rate the previous project outcomes on average based on three success dimensions from 1(one) to 5 (five), whereby 1 is never and 5 represent every time. The type of Likert scale used was measured in frequency of use. For cases where the respondents rated the success dimension using budget, quality and time as "*every time*", the projects are 100% successful. However, if the same success dimensions were rated never or almost never, then that project is regarded as failure. The projects must not exceed time or budget to complete, and must be accepted by the user; this is the meaning of success.

Table 4. 3: *Project Outcome*

Success Dimensions	Measures	Never	Almost never	Occasionally/ Sometimes	Almost every time	Every time	Mean	N
Budget	The project was completed within or below budget	10	26	207	308	200	3.57	209
Quality	The product met the customer's specifications	3	14	102	452	255	3.94	208
	The customer is using the delivered product	3	10	78	312	480	4.23	208
	The customer was satisfied with the project	3	4	96	428	295	4.07	203
Time	The project was completed on time or earlier	14	50	219	268	135	3.33	206

According to the results displayed in **Table 4.3**, the projects are occasionally completed within their original budget and they met the quality specified almost every time. This means that the organisations are measuring the cost of developing software. The projects are sometimes completed in time, since the time success metric was rated 3.33 out of 5. This good quality contributes positively to the success of the software development projects. The Likert scale of 1 and 2 obtained for success dimension time, needs more attention because such projects often experience challenges. As depicted in **Figure 4.3**, the option of "Never" was significantly low;

this simply implies that the previous projects were rated better.

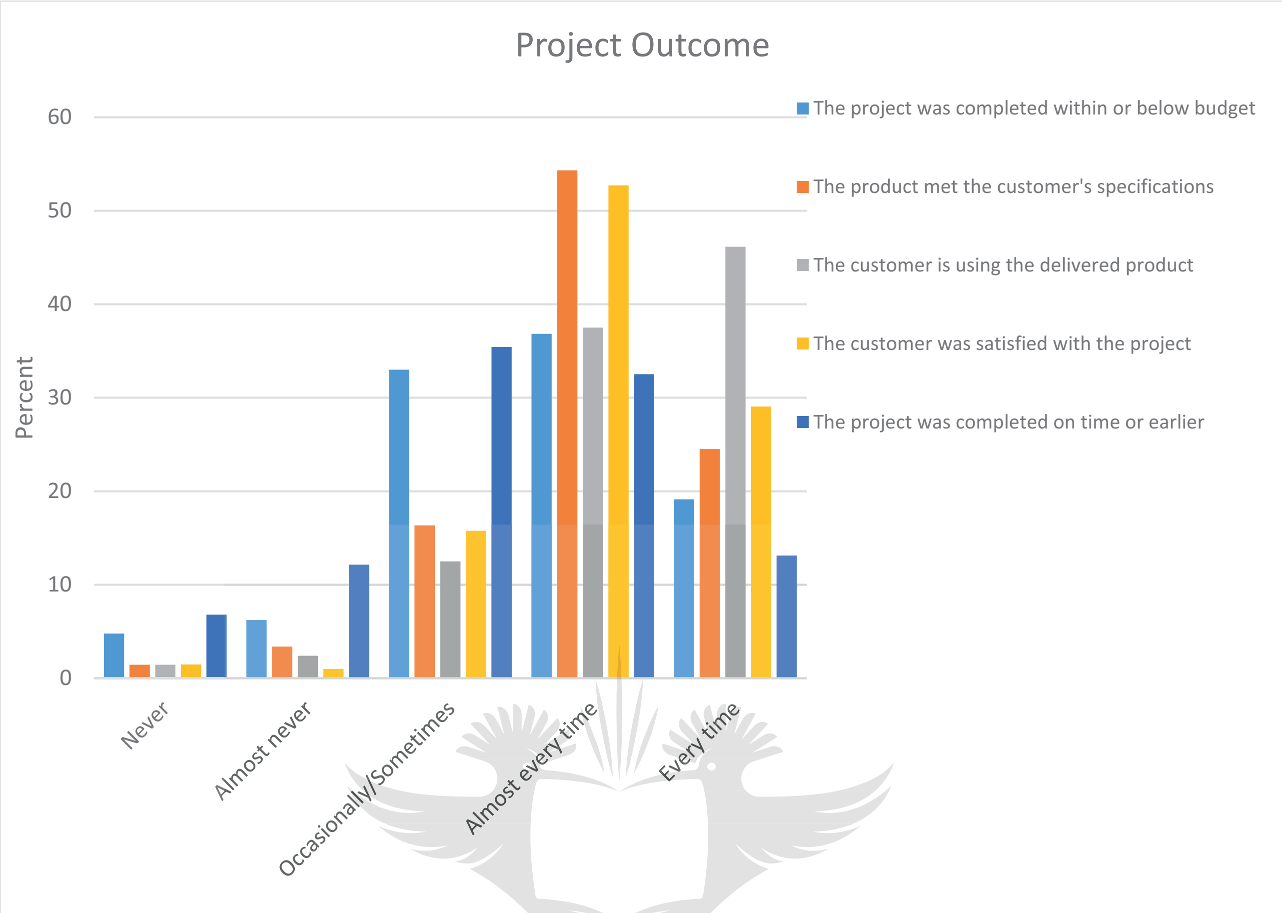


Figure 4. 3: *Project outcome*

The customers are using the delivered product in large numbers, and this is encouraging to the suppliers. Some of the moments of low morale in software development teams is when the project is not functioning as expected. During the last 6 months, the organisations have generally managed to meet the customer's expectations when rated. For example, when the project performance metrics of "*customer is using the delivered product*", "*the customer was satisfied with the project*", and "*the product met the customer's specification*" were assessed, the respective above-average mean values of 4.23, 4.07 and 3.94 were obtained against a 5 point Likert scale. Using the same Likert scale, the project was completed within or below budget (mean is 3.57) and on time (mean is 3.33). However, it is noteworthy that the organisations performed poorly on "*the project was completed on time or earlier*" indicator. The overall perceived performance of 3.886 represent a significant improvement of 77.7%. Surprisingly, a project success rate of 77.7% was also recorded. In recent years, growing negative perceptions about the success rates of project was observed. In this study, significant differences between the successful projects in the South

African IT sector and results reported by the The Standish Group (2014) was observed. A possible explanation can be found in the difference between critical factors of both studies, which is presented in section 4.4.3.4.

VersionOne Inc. has surveyed thousands of software practising professionals and have been providing the requisite software project performance data over the past eleven years (*VERSIONONE.COM*, 2018). The major findings of the VersionOne Inc. study are summarised in **Table 4.4**.

Table 4. 4: *Project Performance Metrics*

Project Performance Metrics	Measures	The state of Agile Report, VersionOne Inc		This Study
		2016	**2018**	**2018**
Budget	The project was completed within or below budget	23%	31%	71.4%
Quality/Scope	The product met the customer's specifications	48%	47%	78.8%
	The customer is using the delivered product			84.6%
	The customer was satisfied with the project	46%	57%	81%
Time	The project was completed on time or earlier	58%	55%	66%

VERSIONONE.COM (2018)/ VersionOne inc. (2018) has recently reported about the state of software agile projects. The business value, on-time delivery of projects and customer or user satisfaction are the top metrics for measuring Agile projects (*VERSIONONE.COM*, 2018). The respondents of VersionOne survey have more available options to measure the rate of their project success, e.g. business value and project visibility as some of their metrics. As shown in **Table 4.4**, the projects of VersionOne are also completed on time as the projects of this study. Respondents of the survey indicated that the customers are using the delivered software products (85%). The respondents of the study need to improve on time it takes to complete projects. When compared with all the other metrics used for assessing project success, time is the one variable that was rated less than 70%.

As mentioned in Chapter 1, other companies that report on the success of software projects include KPMG, PricewaterCooppers and PMI. A comparative analysis of this and the PMI studies, indicate that a higher success rates was obtained in this study relative to the PMI study (2017). A 77% success rate reported in the current study constitutes a significant increase in comparison to the 69% reported by PMI (2017) when the same metrics (budget, quality and time) were considered. In the PMI (2017) study, the organisations surveyed were divided into two categories, namely: Champions and Underperformers. The budget consumption reported in the PMI (2017) study was found to be always higher than 50% (on the average 55%) from the year 2015. The quality performance of the study was divided into three 3 scope variables; a single measure for the same metric was used for the PMI (2017) study. The time taken to complete a project was found to be always closer to 50%. The Champions are defined as organisations with at least 80% of projects being completed on time, budget and meeting the original goals and have business intent and having high benefits realization maturity. The underperformers are defined as organisations which have 60% or less projects completed on time and on budget and meeting original goals and business intent of the organisation and are having low benefits realization maturity. The Champions organisations were found to possess a very high project success rate of 92%. For these Champions organisations, the percentage of projects completed on time and within budget was found to be 88% and 90%, respectively.

Therefore, a 77% success rate for IT projects/software development projects recorded in this research study does not constitute an outlier when compared with other studies. Sauer *et al.* (2007) found that 67% of their projects were delivered close to budget, schedule and scope expectations. The Prosperus Report of 2003 (Sonnekus, Rudi & Labuschagne, 2003:2) reported an IT success rate of 43%. In 2013, established that the success rate of IT projects in SA was increasing and a 59% success rate was reported for the year. The Marnewick's (2013:86) study has also reported an 18% increase in the success rate relative to a previous study conducted in SA.

4.4.3.2 The project size

The respondents were asked what the average size of their projects was focus was on previous projects and on the previous experience of the project respondents. It was found that most of the organisations surveyed (42% of the respondents, i.e. about 90 respondents out of a total of 212 respondents) had been running medium projects (see **Figure 4.5**). At 33.5% (i.e. 70 respondents), large projects occupied the second spot while very large (22 respondents) and small projects) were lagging behind at 10.4% (22 respondents) and 10.8% (23 respondents), respectively. The size of the software product is determined by many factors and can be measured from software specifications (Hastings & Sajeev, 2001). Lines of Code (LoC) (also called source lines of codes) is a common measure of software size (Dolado, 2000; Kenett & Baker, 2010). According to Jones (2013), the size of software can be measured by lines of code, number of requirements, number of classes and functions points.

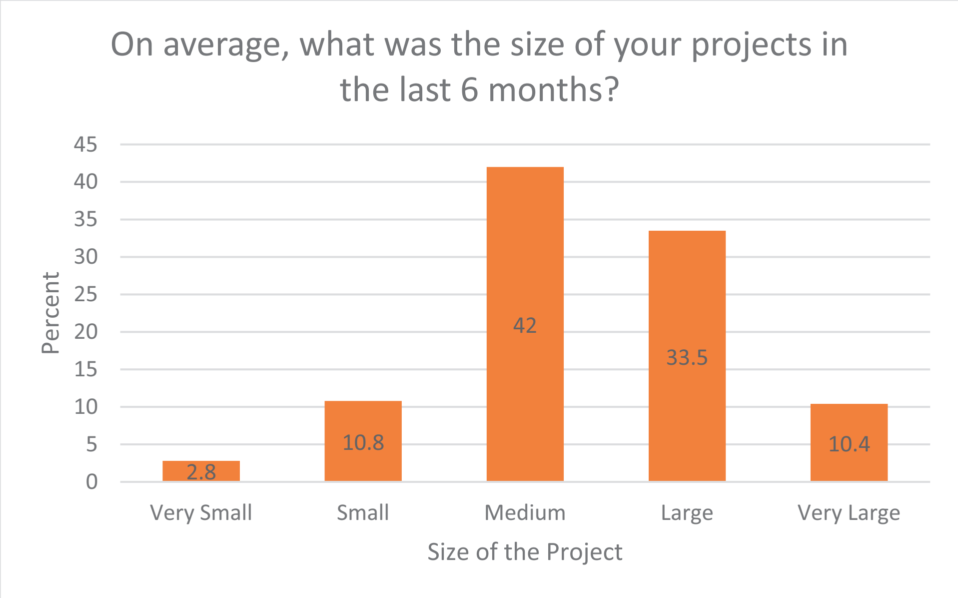


Figure 4. 4: *The Project size*

It was found that most organisations are undertaking medium software development projects, and very small projects, as a result, achieved a very low score. Size of a project is one of the factors used for measuring the quality of the software developed (Kenett & Baker, 2010:122). According to Martin et al. (2007), large information system projects with long duration to complete have more challenges to meet the agreed project budgets and expected quality due to the cost of the technology, high number of staff allocation and hiring of vendors. Sanchez, Terlizzi & de Moraes (2017) have found that challenges such as misunderstanding of specifications, technical problems and testing to be the most unexpected problems that regularly affect the software development of smaller projects.

4.4.3.3 Project involvement

Similar to the previous question (see section 4.4.3.2), this question was directed towards determining the number of projects delivered during last 6 months. The motive of the question was to determine whether or not the software development team had recent projects. Results generated from this study (**see Figure 4.5**) indicate that more than 50% of all the respondents delivered between 1 and 5 projects during the last 6 months. On the other hand, several other respondents indicated that they had 6-15 projects during the same period of time. The average number of projects delivered during the last 6 months was found to be 2.81 for each respondents. The mean value of 2.81 indicates that the organisation surveyed are responsible for multiple projects.

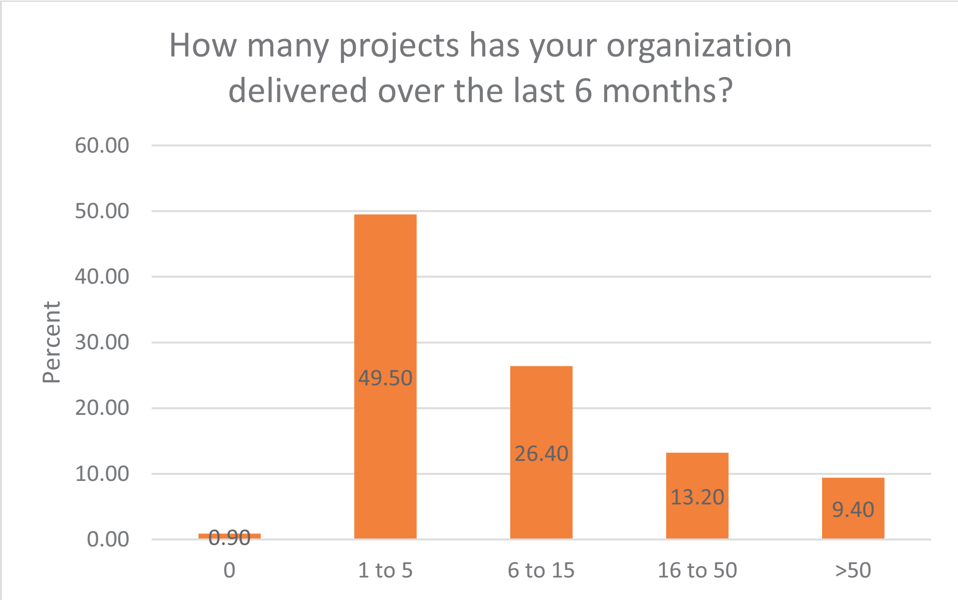


Figure 4. 5: *Project involvement*

In the section that follows, an analysis of the critical factors of software development and how they affect the project outcome is presented.

4.4.3.4 Software development factors analysis

The intention of this question was to establish the factors that critically influence the success of software development project. The questionnaire was composed of a total of 26 software development factors. The structure of factors was divided into three categories, namely: people, processes and technical. Radujković & Sjekavica (2017) regard project success factors as enablers of project success. The 26 critical success factors were sourced from a similar study of Nasir & Sahibuddin (2011), and these factors were measured using a 5-point Likert scale (whereby 1 means strongly disagree and 5 refers to strongly agree). Their research paper included All the possible factors mentioned by Nasir & Sahibuddin (2011) were aligned to software development projects and were adopted for measuring factors that influence the project outcome only.

The selected factors of this research study apply to any software development project. The critical software development success factors by order of criticality for each category are presented in

Table 4. 5: *Critical success factors for software development*

Category name	Rankings	Critical factors	Means	Std. Deviation	N
People related factors	1	Committed and motivated team	**4.52**	0.732	191
	2	User/client involvement	**4.41**	0.816	190
	3	Good leadership	**4.35**	0.789	189

	4	Skilled and sufficient staff	**4.33**	0.793	188
	5	Support from top management	**4.31**	0.811	190
	6	Effective project management skills/methodologies (project manager)	4.25	0.827	191
	7	Good performance by vendors/contractors/consultants	4.22	0.834	189
Process related	1	Clear requirements and specifications	**4.37**	0.868	190
	2	Clear objectives and goals	**4.36**	0.808	191
	3	Proper planning	4.29	0.939	189
	4	Effective communication and feedback	4.28	0.874	190
	5	Clear assignment of roles and responsibilities	4.19	0.820	190
	6	Good quality management	4.18	0.787	188
	7	Appropriate development processes/methodologies (process)	4.15	0.818	190
	8	Adequate resources	4.14	0.876	189
	9	Realistic budget	4.1	0.887	187
	10	Effective change and configuration management	4.09	0.830	190
	11	Realistic schedule	4.07	0.884	191
	12	Up-to-date progress reporting	4.04	0.857	191
	13	Risk management	4.03	0.925	190
	14	Effective monitoring and control	3.99	0.805	191
	15	End-user training provision	3.96	0.913	189
	16	Frozen requirement	3.38	1.035	188
Technical related factors	1	Supporting tools and good infrastructure	4.27	0.746	190
	2	Familiar with technology/development methodology	4.16	0.799	189
	3	Complexity, project size, duration, and number of organisations involved	4.06	0.770	187

The top 10 critical success factors that influence project success are illustrated in **Figure 4.7** by way of a 100% stacked column chart, which show the relative contribution percentage of each critical success factor in stacked columns against the total or cumulative of stacked columns of 100%. was used to compare the percentages that options of each factor contribute to a total. The contribution percentage of the first two options (strongly disagree and disagree) are not shown on the 100% stacked bar chart on **Figure 4.6**, both options contribute less than 6% on each factor and are displayed on the left side of the chart. Out of a total of 212 respondents, only 190 completed the section of critical success factors in the questionnaire. Certain critical success factors received a low rating, frozen requirements (mean = 3.38; SD = 1.035) as shown in table 4.5. Only 3 of the 26 factors that received a rating of less than 4, namely: effective monitoring and control, end-user training provision, frozen requirements, as shown in table 4.5.

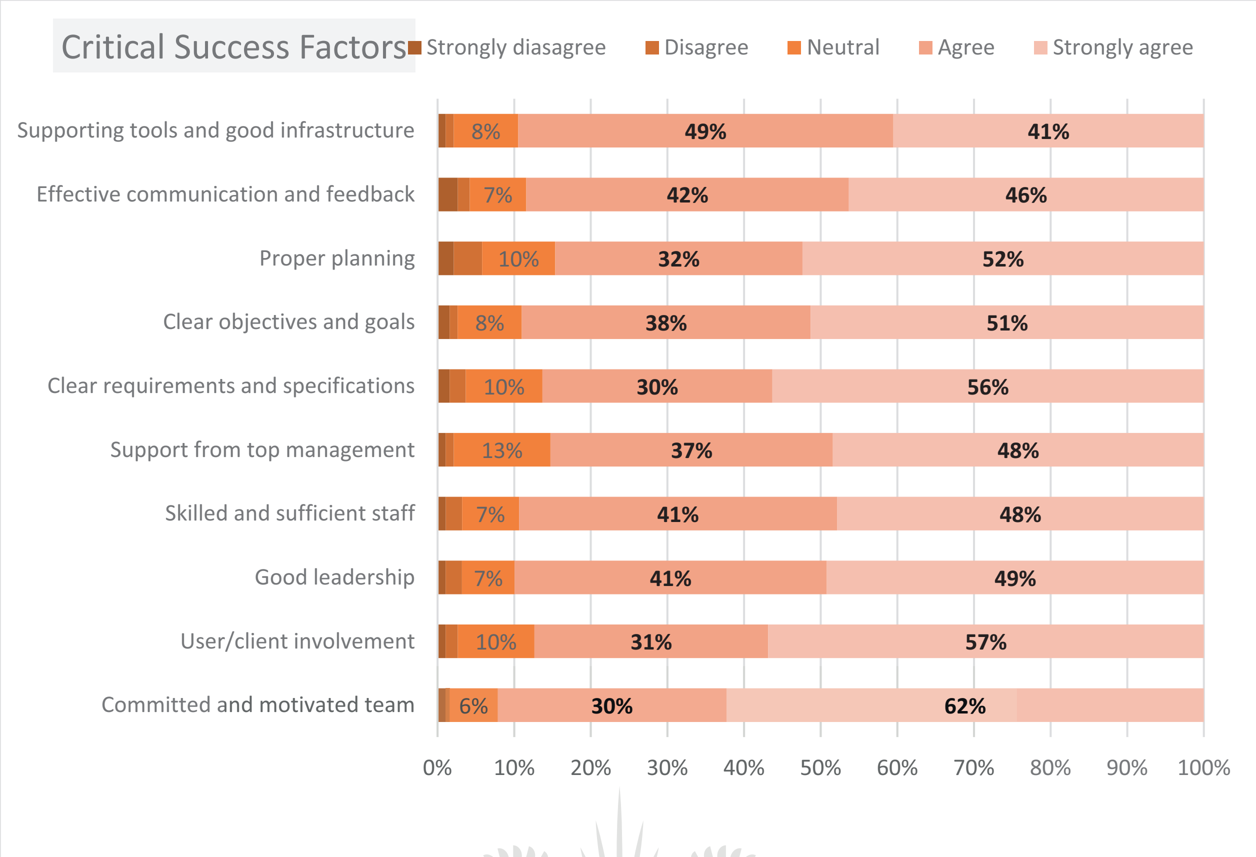


Figure 4. 6: *The top ten critical success factors*

Based on 26 success factors, the Cronbach alpha for the reliability test for this subset is 0.956 and is therefore valid for analysis. The top five factors that are regarded as more critical to the success of software development projects are: committed and motivated team; user/client involvement; clear requirements and specifications; clear objectives and goals; and good leadership. The highest mean score obtained for success factor (committed and motivated team) was 4.52, with a standard deviation of 0.732; this means that the respondents agree that committed and motivated team factor is critical for the success of the software development project. Approximately 62% of the survey respondents indicated that the most important factor for the success of their software development projects is commitment and motivation to their software development team (committed and motivated team).

Three of the top five factors fall under people related categories and only two factors fall under process factors. Based on the findings of this research study, the factor that emerges as the most critical factor is "*committed and motivated team*". According to Gheni, Jusoh & Jabar (2017), the critical success factor of "*committed and motivated team*" is concerned with the interest the team has in the software development project and how the project can be completed within good time, cost, quality and budget. A statistical analysis of the results of this research has revealed that the second most critical factor is "*user/client involvement*", which was respectively ranked 3, 6 and 9 by the CHAOS Report (2015), Nasir & Sahibuddin (2011) and Prosperus Report (2013). When the user/client is involved from the beginning t of the software development project, the development team becomes committed to the project and the requirements become clearly defined.

The third most critical success factor that affect software development is "*clear requirements and specifications*". This factor affects the software requirements and quality of the end-product. Requirements and specifications are regarded by Arias *et al.* (2012) as the main challenges being faced by software development projects. During early stages of software development project, the users do not know exactly what type of product they need.

The fourth most critical success factor is the project must be aligned with the organisation's goals and business objectives, the factor is called "*Clear objectives and goals*". Then amongst top ten critical factors there are other factors that are regarded as the most important, to achieve the project objectives, which is a "*proper project planning".* The project planning needs approval by all the stakeholders. Factors such as "*end-user training provision*"; "*effective monitoring and control*"; "*frozen requirements*" are less critical to the software development projects.

The principal nature of the critical factors that are considered in software industry has not changed that much in the last two decades. As discussed in section 3.7.4.2, a correlation analysis was performed to identify the significant relationships between three software development factor categories (constructs) and software project outcomes. According Lipschutz & Schiller (2012), the correlation coefficient r is derived from a liner relationship and has the following properties:

1) -1 =< r =< 1
2) $r > 0$ if y tends to increase as x increases and $r < 0$ if y decreases as x increases (where x and y are the variables of a linear relationship).
3) An r that is closer to -1 or 1 indicates a strong linear relationship between x and y; an r that is closer to 0 indicates weaker linear relationship between x and y.

The correlation between the project outcome and software critical factors (people, process and technical) was carried out and the correlation results are presented in **Table 4.6**. The project outcome consists of the following statements: the project was completed within or below budget, the product met the customers specifications, the customer is using the delivered product, the customer was satisfied with the project and the project was completed on time or earlier. In terms of the correlation between people related factors and project outcome, people related factors consists of : committed and motivated team; user/client involvement; good leadership; skilled and sufficient staff; support from top management; effective project management skills/methodologies (project manager); and good performance by vendors/contractors/consultants.

Table 4. 6: *Correlation between critical success factors constructs and project outcome constructs*

	Construct	Project Outcome		
		Pearson Correlation	Sig. (2-tailed)	N
Software Development Factors	People	.235**	0,002	173
	Process	.275**	0,000	158
	Technical	.219**	0,004	175

**Correlation is significant at the 0.01 level (2-tailed).

The process related factors are consisted of: clear requirements and specifications; clear objectives and goals; proper planning; effective communication and feedback; clear assignment of roles and responsibilities; good quality management; appropriate development processes/methodologies (process); adequate resources; realistic budget; effective change and configuration management; realistic schedule; up-to-date progress reporting; risk management; effective monitoring and control; end-user training provision; and frozen requirement.

The technical construct is on the other hand made up of only three factors, namely: supporting tools and good infrastructure; familiar with technology/development methodology; and complexity, project size, duration and number of organisations involved.

The respective mean values obtained are indicated in parentheses as follows: people related factors constructs (4.34); process related factors constructs (4.10); technical related factors constructs (4.16), the project outcome constructs was 3.83. Both constructs of three categories of factors (people, process and technical) were used to calculate the correlation person coefficient against project outcome. The project success constructs was made of a total of five items, namely budget, time and the quality (quality is consisted of three measures, which are customer specification, delivered product and satisfaction with the project).

The correlation results revealed that all three constructs have a significant correlation ($p < 0.01$), people (0.002), process (0.000) and technical (0,004). As indicated in **Table 4.6**, there is significant correlation between people, process, technical and project outcome factors since their significant values are less than 1 percent (0.01).

The Pearson correlation coefficients for people, process and technical constructs are 0.235, 0.275 and 0.219 respectively. According to Pallant (2007), this represents a weak relationship and thus suggests that the critical success factors do not contribute significantly to the success of the software development project.

The findings about success factors that are reported in this study compares well with factors previously reported in other studies (i.e. The Standish CHAOS Report (2015), Prosperus Report (2014) and results reported by Nasir & Sahibuddin (2011)).

While both The Standish CHAOS Report (2015) and the Prosperus Report (2014) are longitudinal studies, the study of Nasir & Sahibuddin (2011) is on the other hand cross-sectional. In this research study, the factors used by Nasir & Sahibuddin (2011) were adopted. **Table 4.7** provides a comparative analysis of results relating to success factors contributing to software development projects, which were generated from this research study and the longitudinal and cross-sectional studies mentioned above.

Table 4. 7: *Ranking of Factors contributing to Software Development projects success*

Ranking	Researcher observations	Nasir and Sahibuddin (2011)	CHAOS Report (The Standish Group, 2015; Hastie & Wojewoda, 2015)	Prosperus Report (Marnewick, 2013) & Joseph & Marnewick (2014)
1	Committed and motivated team	Clear requirements and specifications	Executive support	Requirements definition clarity
2	User/client involvement	Clear objectives and goals	Emotional maturity	Communication between team and customers
3	Clear requirements and specifications	Realistic schedule	User involvement	Communication between project team members
4	Clear objectives and goals	Effective project management skills/methodologies	Optimization	Business objectives clarity
5	Good leadership	Support from top management	Skilled resources	Understanding of users' needs
6	Skilled and sufficient staff	User/client involvement	Standard architecture	Project manager competency
7	Support from top management	Effective communication and feedback	Agile Process	Executive support
8	Proper planning	Realistic budget	Modest Execution	Handling change
9	Effective communication and feedback	Skilled and sufficient staffs	Project management expertise	User involvement
10	10.Supporting tools and good infrastructure	Frozen requirement	Clear Business Objectives	Change control processes

According to the data displayed in **Table 4.7**, the ranking of the project success factors of this research study is surprisingly different from those of the other above-mentioned studies. In this research study, the top four success factors that influence software development projects are ranked in the following order from highest to lowest: committed and motivated team; user/client involvement; clear requirements and specifications; and clear goals and objectives. As far as the finding of this research study are concerned, the software development team must remain motivated and committed, otherwise the software project will suffer. Experts regard individual and team motivation as the leading success factor that affects the productivity of a project team (Motivation in Project Management, not dated). The predecessor study of Nasir & Sahibuddin (2011) used the same 26 critical success factors adopted by the research study. The top six critical success factors of Nasir & Sahibuddin (2011) are two process related factors, which are ranked two (clear requirements and specifications) and four (clear objectives and goals) as well as one people related factor that is ranked number two (user/client involvement). This research study and Nasir & Sahibuddin (2011) have found different rank 1 and 2 critical success factors of software projects.

Too many factors emerge when the two longitudinal studies reported in the CHAOS and Prosperus reports are scrutinized. The top four critical success factors for the delivery of software projects, which were reported in the CHAOS Report, differ slightly from those reported in this research study. This research study and The Standish CHAOS Report share only a single critical success factor (i.e. user involvement), which is ranked second and third in this research study and the CHAOS Report.

When The Standish Group surveyed IT executive managers about the success factors that influence their projects, four main factors were mentioned, namely: executive support, emotional maturity, user involvement and optimization. Three of the factors mentioned by these executives are not in agreement with those found by this research study. According to the CHAOS Report, the two top software development projects factors that have remained unchanged for the past five years are (their rankings in parentheses): executive support (1); and user involvement (2). Executive support continues to dominate the top influencers of project success. Similar to the study by Nasir & Sahibuddin (2011), the Prosperus Report and the study of Joseph & Marnewick (2014) do not differ much with study results obtained in this study. Our results and those of the above-mentioned studies share the following critical success factors: requirements definition clarity; and communication between team and customers; communication between project team members; business objectives clarity. Although the requirements are not regarded as critical factors by the CHAOS Report, they continue to dominate the area of project success as the most important critical factor; this was confirmed by the Prosperus Report (2013), Nasir & Sahibuddin (2011) and the current study (ranked number 1 by both Nasir & Sahibuddin (2011) and Prosperus (2013), and number 3 in this research study). Furthermore and with regards to specifications, Arias *et al. (*2012) have asserted that planning and other estimates will also become invalid if the requirements are not clear.

4.4.4 MATURITY BROKEN DOWN

The scale between 1 and 5 determines whether the software organisation is immature or not. As discussed in Chapter two, immature organisations spend more time reacting to crises, while the software processes of matured organisations are consistent and follow a set of disciplined processes throughout the software project (Marchewka, 2013). This section presents the overall results on maturity level. The maturity is broken down at software team, project management and organisational levels. This means that the analysis was done from team to organisation levels. Each level has a table with analysis that includes mean values, number of respondents that completed the question and standard deviations. The maturity levels were calculated using a 1 to 5 scale of maturity level; 1 indicates low level of maturity and 5 indicates a very high level of maturity.

4.4.4.1 Software development process maturity measures

For the software development maturity, the following six constructs were used: requirement management, requirements development, technical solution, product integration, verification and validation. The respondents were asked to indicate to what degree they agree that the process was implemented within their software development process. The Cronbach alpha for this subset is 0.963 (*number of items =25*), and is hence valid for the analysis. The focus here was on software development processes than organisation level. One of the processes of software development process, is for example, defined in CMMI_dev as: software requirement management, which is the management of technical and non-technical requirements generated by the project or work group (Chemuturi, 2013).

The requirements can be generated from project by the software development team or any stakeholder such as external source as customer. The process includes planning, organising, staffing and controlling, and there are many agencies that are responsible for the management and classification of the requirements (Chemuturi, 2013).

Table 4.8 indicate an average maturity level of 3.95 out of 5 for software development. With a mean level of maturity of 3.69, the specific goal (objective) "*product or product component solutions are selected from alternative solutions"*, which falls under process area called technical solutions, received the lowest level of maturity. On the other hand, a final test environment was found to be as close as possible to the environment in which the product or product components will be used in real system life.

Table 4. 8: *Software Development Process Areas Constructs*

Specific Goals/ Activities	N	Mean	Std. Deviation
A final test environment is close as possible to the environment in which the product or product components will be used/performing life.	173	4.18	0.907
The software team ensures the product meets its specified requirements.	176	4.14	0.805
The work products that needs verification are identified.	174	4.06	0.881
Software product components designed are implemented.	175	4.06	0.712
The success of the integration is validated.	176	4.06	0.836
The customer requirements are refined and elaborated to develop software product and its component required.	178	4.01	0.850
Packaging the assembled products components for the delivery to customer.	173	4.01	0.846
The software development team make sure that the assembled product components are ready for integration.	176	4.01	0.872
Preparation for product integration is conducted.	175	4.00	0.837
Establish verification procedures and criteria.	174	3.99	0.877
Selected work products are verified against their specified requirements.	174	3.98	0.915
Product or product component designs are developed.	176	3.96	0.781
The product integration environment is prepared.	175	3.94	0.814
Preparation for validation is conducted.	177	3.94	0.887
The product component interfaces are tested (both internal and external) for compatibility before starting with the integration activities.	175	3.94	0.923
The requirements are analysed and validated against risks in the early phases of the software projects.	177	3.92	1.014
The stakeholder needs, expectations and interfaces are translated into customer requirements exactly as required.	176	3.91	0.925
Appropriate verification environment is prepared.	175	3.91	0.896
The commitment to requirements is obtained from the project participants.	178	3.90	0.927
Supporting documentation are implemented from their designs.	177	3.88	0.973
During customer requirements development, the stakeholder needs, expectations and interfaces are collected by software development team.	177	3.85	1.008
Product integration sequence is in place.	176	3.83	0.878
Peer reviews are performed on selected work products to ensure it meets specified requirements.	174	3.83	0.982
Project participants manages the changes imposed to existing requirements.	177	3.79	0.951
Product or product component solutions are selected from alternative solutions.	178	3.69	0.975

Figure 4.7 provides general software development maturity levels. The respondents were not asked to indicate the level that they believe their software development team belongs to. However, the respondents were asked questions that determines the level of the maturity when answered. The perceived levels of software development maturity are based on process area and its specific practice goals and objectives of the processes. Overall, there is strong dominance of level 4, where almost 50% of respondents indicated that their software development team perceive maturity level 4, which means processes are controlled using statistics and quantitative techniques. About two-third of the respondents are classified, in maturity levels 4 and 5, while a smaller fraction is classified in maturity levels 1, 2 and 3.

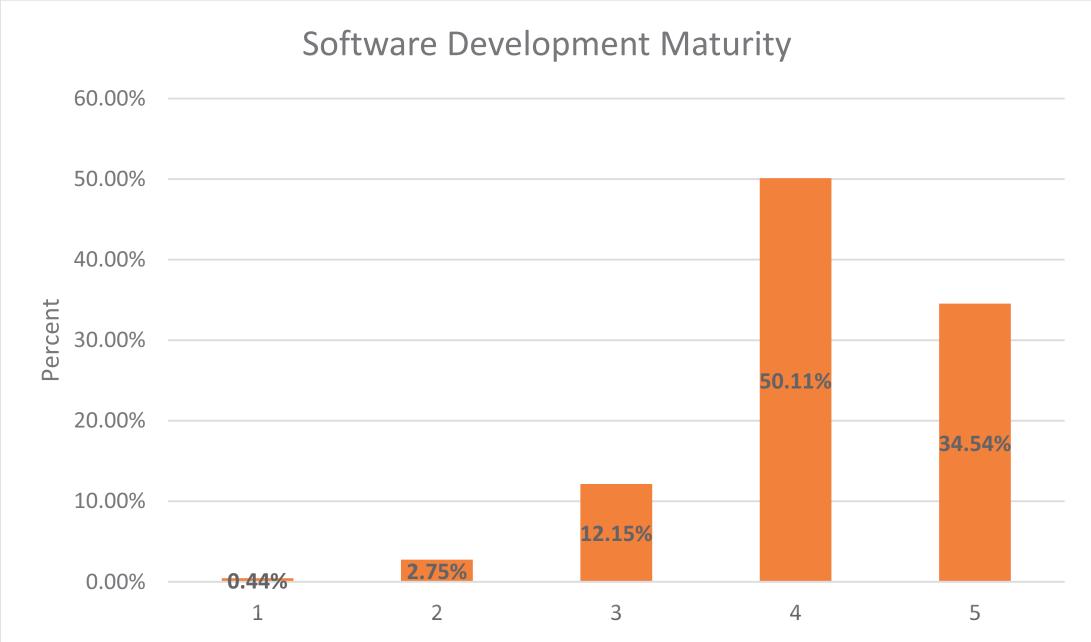


Figure 4. 7: *Perceived levels of software development maturity*

The maturity level of software development processes is illustrated in **Figure 4.8**. Most software development team members agree that they follow a validation process for their software development, and some experienced challenges with managing the requirements. To this end, requirements management received a low maturity level of 3.85, which is the minimum of all the process areas of software development processes. This means the respondents are perfect in validation (4.09) but need improvement on requirements management and delivering technical solution. After software development dimension, software development project has another main activity dimension that deals with proper planning and controlling of the development activities to meet project goals with regards to budget, time and quality (Jalote, 2002:3).

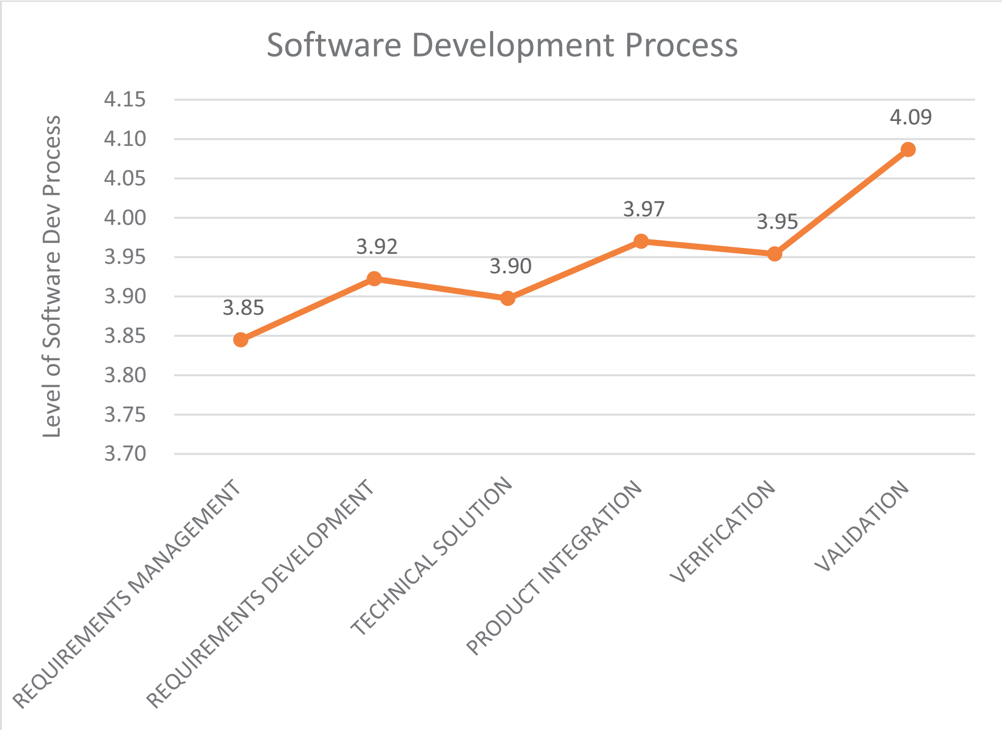


Figure 4. 8: *The Results of The Software Development Maturity assessments*

4.4.4.2 Project management maturity measures

Some of the organisations rely on project management as their core business. The Cronbach alpha for this subset was found to be 0.965 (*number of items = 19);* this is valid for the analysis. The focus here was on project management processes than organisation level.

The questionnaire had 19 statements that address the elements of project management maturity. The statements dealt with project planning, project monitoring and control, supplier agreement, integrated project management, risk management and quantitative project management. The respondents were asked to rate/indicate to what extent do they agree that the process was implemented within their organisation. A rating scale of 1-5 was used (1 means strongly disagree, while 5 means strongly agree).

As shown in **Table 4.9**, the organisations managed to monitor the delivery of their product, and both their project and suppliers are satisfied with the agreements that were in place. Quantitative project management construct is the one process area that require more attention.

Table 4. 9: *Project Management Process Areas Constructs*

Descriptive Statistics				
Specific Goals/Activities	N	Sum	Mean	Std. Deviation
A project plan is established as the basis for managing the project (e.g. plan for data management, needed knowledge and required skills).	169	689	4.08	.880
The creation and delivery of the product is monitored.	162	656	4.05	.794
Commitments by stakeholders to the project plan is obtained.	169	679	4.02	.942
The project is conducted using a defined process tailored from the organisation's set of standard processes.	168	674	4.01	.869
The collaboration between the project and relevant stakeholders is emphasized.	167	669	4.01	.882
Both the project and suppliers are satisfied with the agreements.	165	658	3.99	.796
Actual project progress and performance against the project plan are monitored.	171	678	3.96	.907
The estimates of project planning parameters (scope of the project, work, effort and cost required) are established.	171	674	3.94	.925
Corrective actions are managed to closure when the project's performance or results deviate significantly from the plan.	171	674	3.94	.962
The vendors qualified to supply the required types of products or product components are determined.	165	649	3.93	.813
The coordination issues that might arise between relevant stakeholders and project teams are resolved.	166	651	3.92	.881
Agreements with the suppliers is established according to the types of acquisitions made.	165	646	3.92	.807
A shared vision of the project is always ensured among individual teams.	163	629	3.86	.987
A risk management strategy is in place to categorise typical and known risks.	169	650	3.85	.926
The risks identified are analysed to determine their relative importance.	168	644	3.83	.958
Risks are handled and mitigated as appropriate to reduce adverse impacts on achieving the objectives.	166	634	3.82	.999
The process performance are managed in order to remain in line with the project objectives.	169	639	3.78	.991
The selected sub-process performance of the project are measured and their results are analysed.	170	638	3.75	.984
Preparation for quantitative management is conducted by establishing performance objectives.	168	612	3.64	1.022

Figure 4.7 illustrates the perceived project management maturity levels of the respondents. In this study, 70% 168/212) respondents and nearly 48% of respondents indicated that their project management maturity level is 4, and 26% respondents think that their maturity level is 5. The study found higher percentage on level 4, which means in overall the project management maturity of the respondents of the study is 3.91.

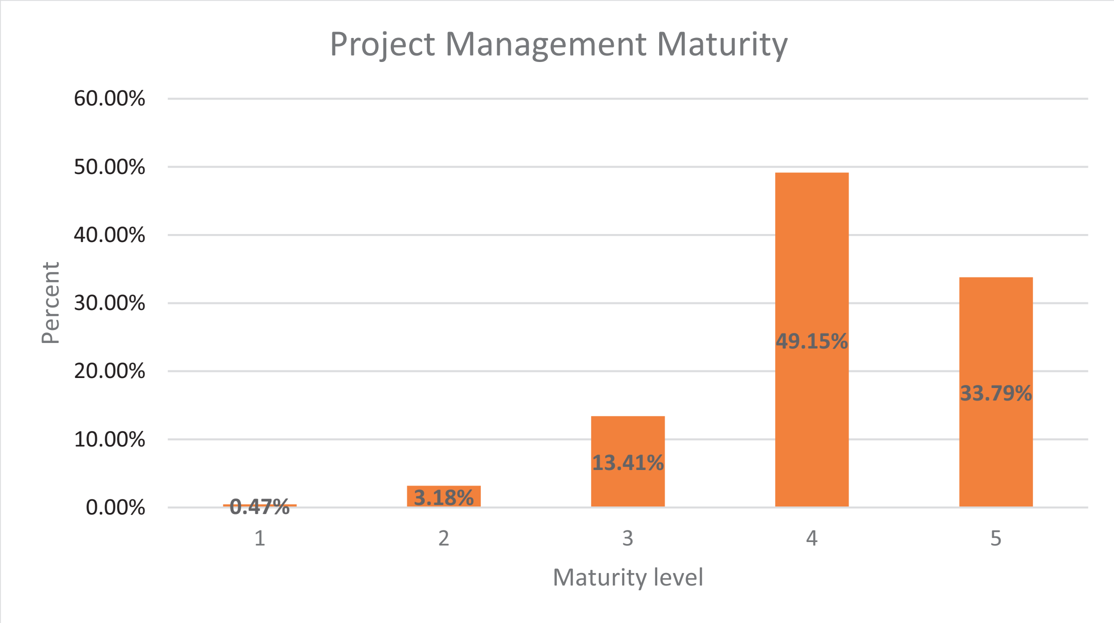


Figure 4. 9: *Perceived maturity levels of project management maturity*

Results for the assessment of project management maturity (see **Figure 4.10**) have revealed that the organisations of the respondents are planning their projects well (maturity level 4.01). However, other project management processes such as quantitative project management (project management maturity level 3.72) still requires further improvement. A quantitative project management process was rated maturity level of 3.7 by the respondents, which is the only lower level of maturity amongst project management processes.

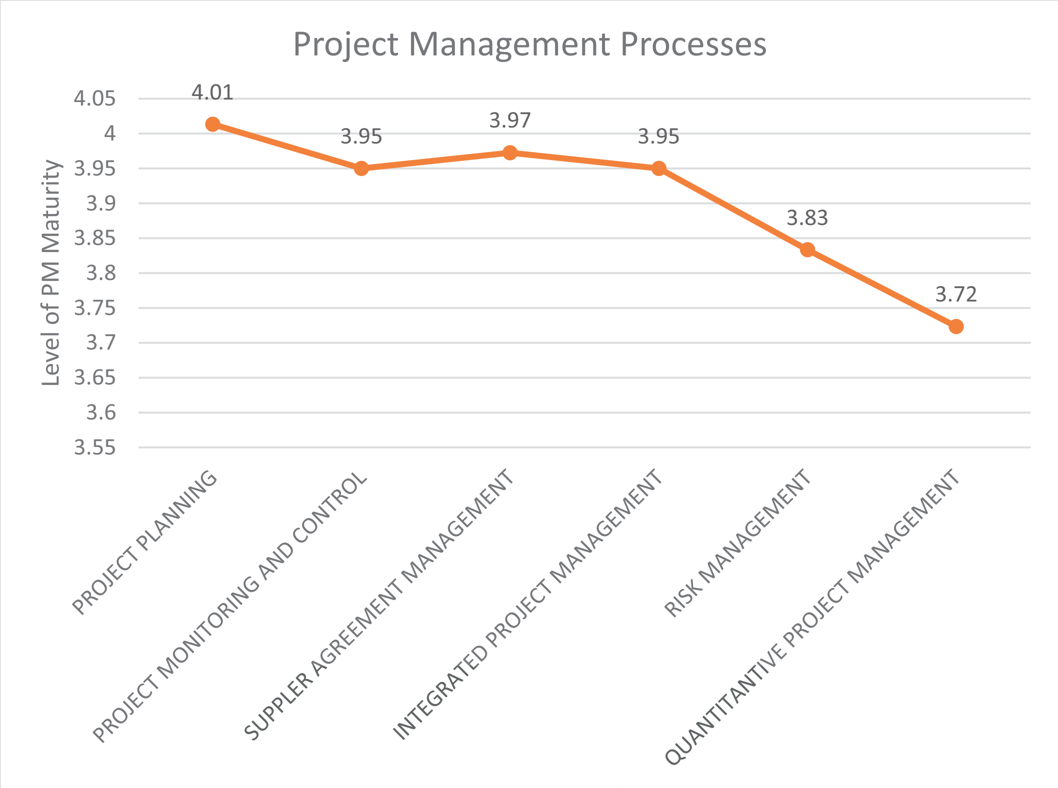


Figure 4. 10: *The Results of project management maturity assessments*

4.4.4.3 Organisational maturity measures

The section of maturity level on organisation level consists of 21 questions from 5 process areas. The Cronbach alpha for this subset is 0.971(number of items = 21), which, like the software development and project management maturity processes areas, is also valid for analysis. The organisation can tests itself and take actions to compare itself and other organisations in the same and different cultures (Andersen & Jessen, 2007). According to the PM Solution (2014) report, the high performing organisations are those that are much more mature in their project management practices when compared with low performers and their maturity level ranges from 3.4 to above.

Respondents rated their organisations processes on a scale of 1 to 5, where 1 is strongly disagree and 5 is strongly agree.

As shown in **Table 4.10**, organizational process performance is one process/construct that needs improvement, along with organizational performance management, both of which have received the lowest mean values (3.65 and 3.8, respectively). It seems that the organisations have managed to implement process improvement over time, and they planned their deployments as required by the maturity model adopted.

Table 4. 10: *Organisational Maturity Constructs*

Descriptive Statistics				
Specific Goals/ Activities	N	Sum	Mean	Std. Deviation
The organisation establishes a standardized set of processes that teams within your organisation can access.	171	681	3.98	.871
The organisation implement process improvements over time.	169	663	3.92	.880
The organisation identify process improvements targets.	169	662	3.92	.922
The deployments are planned.	170	664	3.91	.937
The organisation develops skills and knowledge for its employees by offering the training.	170	657	3.86	.967
The organisation determines the process improvement opportunities that it will focus on (e.g. Improvements needs of software used).	170	657	3.86	.935
A training capability is developed (resources and materials), which supports the organisation to deliver the courses.	170	657	3.86	1.014
The organisation analyses the improvements opportunities and proposals.	171	660	3.86	.948
The organisation has a plan to establish process action plans.	169	652	3.86	.888
The organisation assesses the process improvements needs.	170	653	3.84	.925
The organisation shapes standardised set of processes so that they can manage their work.	168	645	3.84	.911
The organisation measures improvements results.	168	643	3.83	.960
Your organisation systematically deploys the improvements into the organisation.	170	648	3.81	.961
The improvements deployed are managed against the plan.	170	644	3.79	.980
Organisational process assets are deployed across the organisation.	168	632	3.76	.949
The organisation collects feedback on training provided.	168	632	3.76	1.034
Process related experiences are incorporated into organizational process assets.	165	618	3.75	.935
The promising improvements are chosen for deployments.	170	632	3.72	.956
The organisations is piloting the selected improvements.	171	634	3.71	.956
The performance measuring mechanics is established.	167	612	3.66	1.056
The performance of the selected processes is recorded.	167	606	3.63	1.073

The perceived organisational maturity levels are presented in **Figure 4.10**. About 169 respondents completed the section of the questionnaire relating to perceived organisational maturity. The question response rate was 80% (169 respondents), % with 0.65% of the respondents indicating that their organisational maturity level is 1 (i.e. initial level), 3.85% indicating an organisational maturity level of 2 (repeatable), 15.56% indicating an organisation maturity level of 3 (defined), 49 % indicating an organisation maturity level of 4 (managed process), and 30.88% indicating an organisation maturity level of 5 (optimum). Optimum as the most matured level. The highest perceived level of organisational maturity is at level 4 (49%) followed by level 5 (31%). The overall perceived organisational maturity level was found to be 3.89.

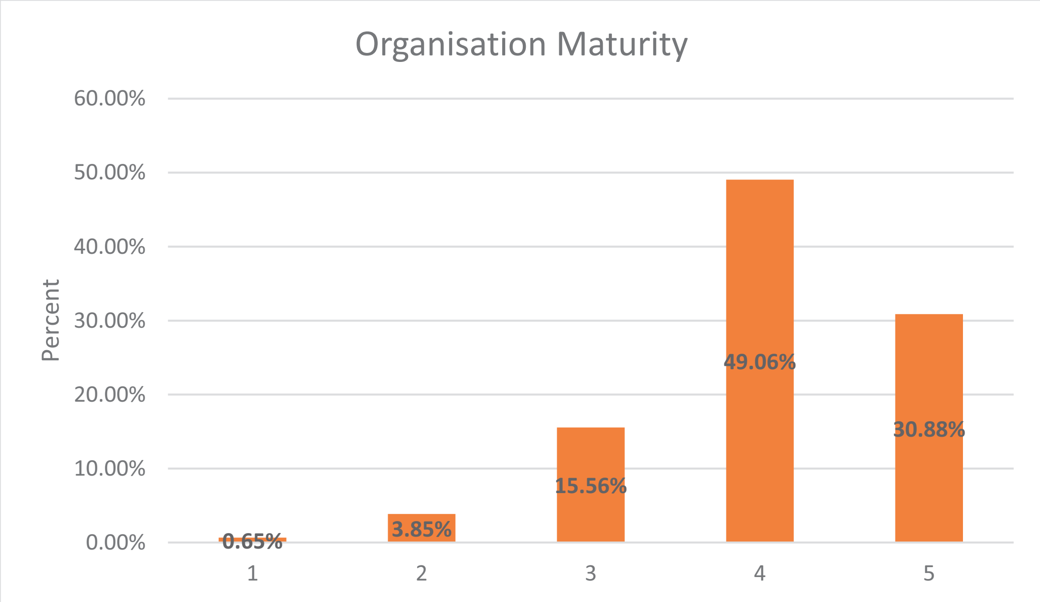


Figure 4. 11: *Perceived levels of organisation maturity*

A comparative analysis of the maturity levels across the five organisational processes (see **Figure 4.12**) indicate that the majority of respondents are following organisational processes as defined by CMMI. The two process areas with the lowest average maturity level are organizational process performance (3.65) and organisational performance management (3.8).

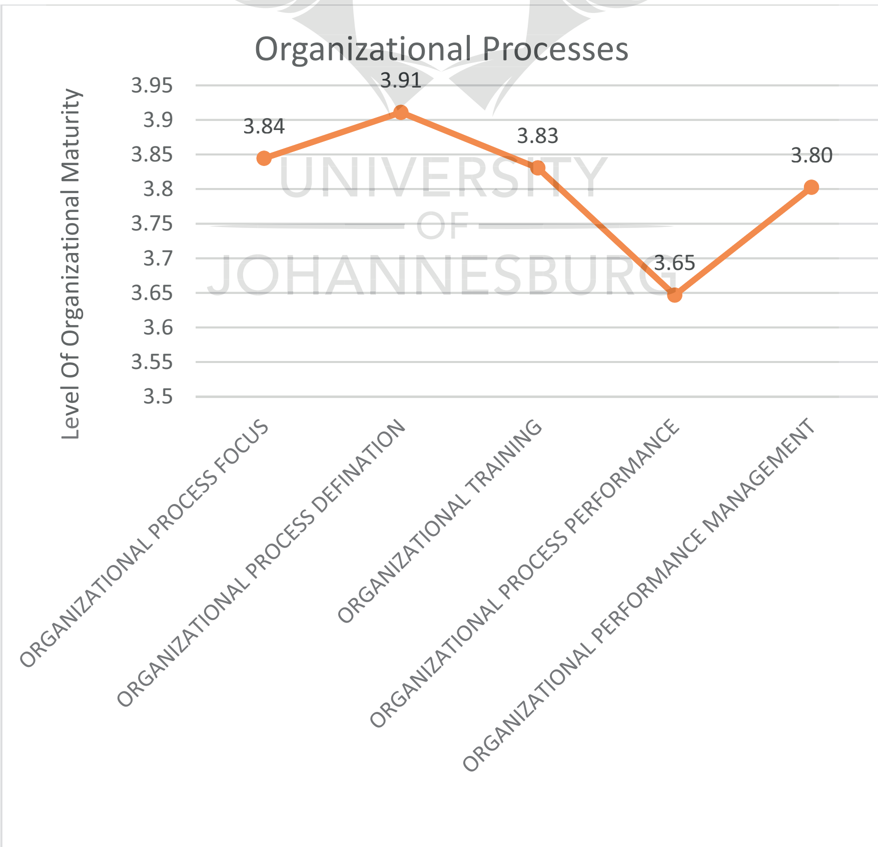


Figure 4. 12: *The results of the Organizational Maturity assessments*

In order to obtain the overall maturity level, the overall average mean of all the three combined means of maturity levels as calculated as follows 3.95 + 3.91 + 3.81 = 11.67/3=3.89 (i.e. the mean of the overall maturity levels of software development maturity, project management maturity and organisational maturity). As shown in **Figure 4.13**, all the three overall maturity levels are dominant at level 4. The maturity level of the ICT industry of 3.1, which was determined by Marnewick (2013a) using PMMM, is lower than the maturity level of 3.95 obtained for this research study.

The maturity level of the study can be converted from 3.89 to 4, while Marnewick (2013a) can be regarded as maturity level 3 since it is 3.1.

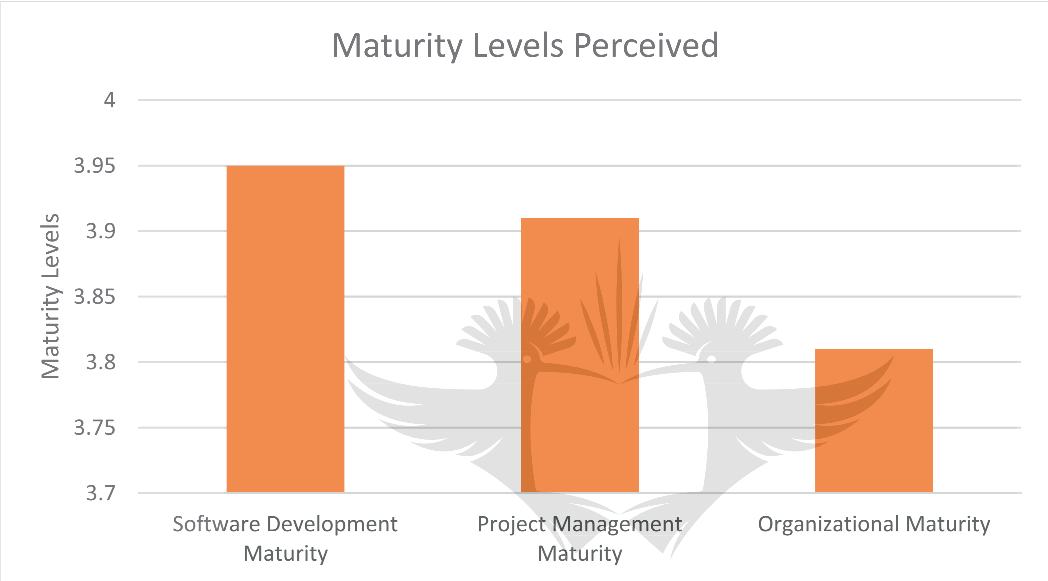


Figure 4. 13: *Overall Maturity levels*

A comparison of the maturity levels of IS projects from Marnewick (2013:13) and the perceived maturity level of this research study is presented in **Figure 4.14**.

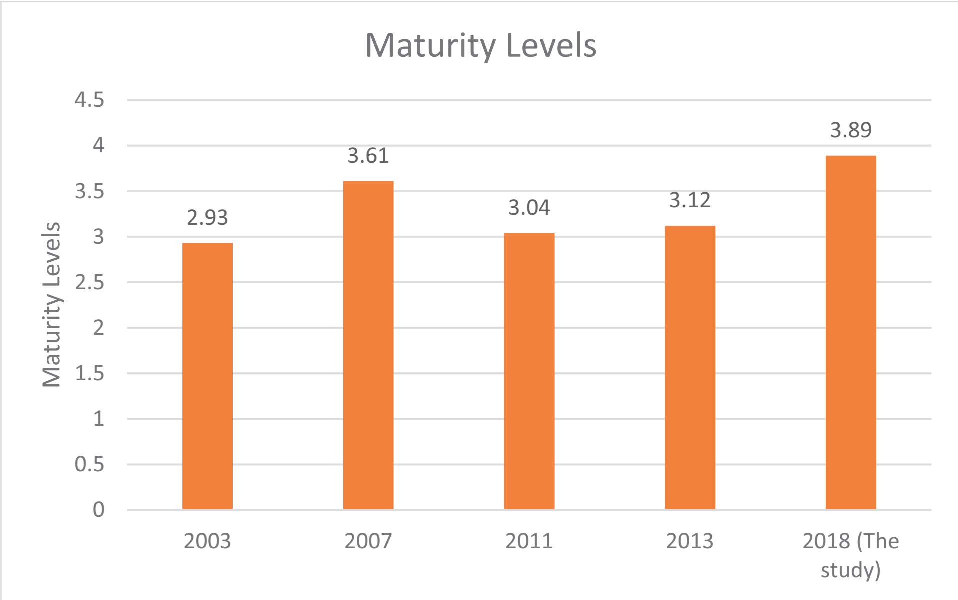


Figure 4. 14: *A comparison of the maturity levels of IS projects (Marnewick, 2013:13) with perceived maturity level of this research study*

Prior to the undertaking of this research study, the highest maturity level of 3.61 was recorded in 2007. Maturity level figures for the current study show an improvement to 3.89. Whereas, the maturity level figures of 2003 to 2013 were more or less the same (save for 2007), the lowest maturity level was observed in 2003 (maturity level = 2.93). Marnewick (2013) has attributed the lower maturity level figure for 2013 to the low number of respondents. This also applies to this research study whereby out of 212 respondents only 170 completed the maturity questions; the maturity section of the questionnaire was left blank by the rest (42) of the respondents. According to Marnewick (2013), only 220 respondents participated in the 2003 study with the lowest maturity level of 2.93; this does not differ much from the 170 respondents of the this study. Overall, the maturity level in South Africa is between 3 and 4. According to Marnewick (2013), such maturity level figures indicate that most projects in the country are functioning at a level where the processes are defined and approved processes are followed. The following section 4.4.4.4 relates the maturity processes and the project outcome.

4.4.4.4 The correlation between maturity measures and project outcome

The main purpose of statistical analysis is to identify the existence of a relationship between independent and dependant variables (Walliman, 2005:305). According to Pallant (2007) and Woodwell (2014:38), the correlation is a statistical technique that is used to describe the strength and direction of the relationship between variables. **Table 4.11** depicts the strength of the relationship between project outcome and maturity. The project success construct was made up of five items, namely: budget and time and the three measures of quality (i.e. customer specification, delivered product and satisfaction with the project). The maturity measures encompass software development process, project management and organisation. The specific goals of each process area were grouped and averaged under a single construct and compared against the project outcome. The project outcome is also consisted of the construct (average) of three project success dimensions (budget, quality and time). The specific goals can be regarded as sub-scales that belongs to process areas, and process area can be regarded as a scale.

Table 4. 11: *The correlation between project outcome and maturity level (constructs)*

Maturity Of	Process Area (Construct)	Specific Goal (Variables)	Project Outcome		
			Pearson Correlation	Sig. (2-tailed)	*N*
Software Development Process	Requirement Management	C1.1, C1.2	.196*	0.012	164
	Requirement Development	C2.1, C2.2, C2.3, C2.4	.352**	0.000	162
	Technical Solution	C3.1, C3.2, C3.3, C3.4	.366**	0.000	162
	Product Integration	C4.1, C4.2, C4.3, C4.4, C4.5, C4.6, C4.7	.383**	0.000	159
	Verification	C5.1, C5.2, C5.3, C5.4, C5.5	.231**	0.003	162
	Validation	C6.1, C6.2, C6.3	.263**	0.001	163
Project Management Maturity Level	Project Planning	D1.1, D1.2, D1.3	.338**	0.000	156
	Project monitoring and control	D2.1, D2.2	.370**	0.000	159
	Supplier agreement management	D3.1, D3.2, D3.3, D3.4	.315**	0.000	153
	Integrated project management	D4.1, D4.2, D4.3, D4.4	.412**	0.000	151
	Risk management	D5.1, D5.2, D5.3	.212**	0.009	151
	Quantitative project management	D6.1, D6.2, D6.3	.429**	0.000	157
Organisation level: Process Management Question	Organisation process	E1.1, E1.2 E1.3 E1.4 E1.5 E1.6 E1.7	.354**	0.000	145
	Organisation process definition	E2.1, E2.2	.369**	0.000	158
	Organisation training	E3.1, E3.2, E3.3	.355**	0.000	157
	Organisation process performance	E4.1, E4.2	.450**	0.000	154
	Organisational performance management	E5.1, E5.2, E5.3, E5.4, E5.5, E5.6, E5.7	.402**	0.000	155

***. Correlation is significant at the 0.01 level (2-tailed).*
**. Correlation is significant at the 0.05 level (2-tailed).*

The first correlation between software development processes and project outcome was done to determine the strength and direction of the relation between project outcome and software development maturity levels (see **Table 4.11**).

A medium positive statistical significant correlation between project performance and each maturity type, r(212)=0,3 ; p< 0.001. The overall coefficient is between 0.196 and 0.450, which according to Pallant (2007) represents a medium relationship of variables.

As indicated by Pearson's correlation coefficient, relationships between project success and maturity model were found to exist. In general, a medium relationship between maturity levels and the project outcome was observed.

Table 4.11 indicate the existence of a strong positive relationship between the variables as well as a significant association at p-value = 0.001 < alpha, alpha = 0.005. The correlation coefficient between maturity models and project outcome is closer to 0.4. For the constructs of software development maturity model, the Pearson value of 0.383 found for product integration represents a medium relationship, and both the respective validation and verification of 0.231 and 0.263 indicate a small or weak relationship.

With a correlation coefficient of 0.196, the requirement management construct also has a weak relationship with project success. The correlation between project management maturity and project success was presented by six constructs. The constructs which has a strong medium relationship is the *quantitative project management* (0.429). The relationship between IT Project Management Maturity and Project success was previously reported by Labuschagne, Jakovljevic & Marnewick (2009) (The Prosperus Report), and no significant correlation was found to determine if the project success is dependent on the maturity level of an organisation. The level of Pearson correlation of 0.273** and the significant level of 0.01 (1-tailed) suggest the existence of a weak relationship. Relative to the study by Marnewick (2013), an improvement in the correlation was noted (see **Table 4.12**).

Table 4. 12: *Correlation between overall maturity management and project success (Sourced from (Marnewick, 2013)*

	2003	2007	2011	2013
Pearson correlation	0.094**	0.08	0.211**	0.207**
Sig. (2-tailed)	0.008	0.28	0.00	0.001

From 2003 to 2013, no significant relationship between project management maturity and project success was noted. As shown in **Table 4.12**, the Pearson correlation coefficient was small although in 2013 it had improved to 0.207**. The relationship was found to be weak, even though a p-value that is below 0.05 was recorded.

The study has found improved medium strength relationship of above 0.30** on many process areas. These results indicate that overall each maturity process has a medium strength of relationship. Requirement management and verification processes of software development maturity and risk management process of project management maturity are the only three processes that depicted small strength relationship with project outcome. Similar to results reported by Price Water Coopers (2004), this research study has found that in most cases higher maturity model organisations deliver superior performance projects and business benefits. Furthermore, the relationship between maturity level and project outcome are significant since the significant levels are below 0.01. The recent report of 'PMI's Pulse of the Profession ™ The High Cost of Low Performance Pulse Perspective' (2018), has released the current state of global project management performance and the value delivery of capability maturity that leads to greater project performance. According to the PMI (2018) study, higher maturity organisations completed 64% of their projects on time, whereas low maturity organisations completed only 36% of their projects on time. Also, while higher maturity organisations completed 67% of their projects within budget, low maturity organisations completed 43% of their projects within budget. The PMI (2018) study is one of the studies that emphasise a great need for achieving a higher maturity level.

4.5 CONCLUSION

Several questionnaires were distributed to the target population, and a good return rate was of 28%was achieved and the response was found to be usable. Upon request by the respondents, the link to the survey instrument was e-mailed and an electronic MS Word version of the survey and ethical clearance were attached to the e-mails. Some of the respondents requested hard copies. As shown in **Table 4.1**, the Cronbach's alpha value that was determined using reliability statistics suggests a good internal consistency for the scales, and values above 0.8 were hence preferable.

The data was presented and the findings from the study are intended to be shared with industry experts and the software industry at large. The analysis of data started with presentation of reliability and validity in order to clarity the reliability and consistency of the results. Thereafter, the respondents' background was presented. It was revealed that most active respondents have work experience of 1 to 5 years, followed by 5 to 10 years. It suffices to say that the two categories of work experience are dominated by software developers.

Table 14.13 represent the country's overall IT spending, including software development projects, for the period 2017 to 2019.

Table 4. 13: *IT Spending Forecast, South Africa (Millions of Rand)*

	2017		2018		2019	
	Spending (million R)	Growth (%)	Spending (million R)	Growth (%)	Spending (million R)	Growth (%)
Data Center Systems	7,803	-3.6	8,594	10.1	8,456	-1.6
Software	27,908	12.7	31,396	12.5	35,361	12.6
Devices	39,634	3.0	39,995	0.9	44,196	10.5
IT Services	71,942	8.7	77,672	8.0	83,303	7.2
Communications Services	117,777	0.8	118,929	1.0	119,361	0.4
Overall IT	**265,065**	**4.2**	**276,586**	**4.3**	**290,677**	**5.1**

Source: Gartner (August 2018)

As indicated in **Table 4:13**, South Africa is investing hugely on IT, although the benefits are growing at the very slow pace. Software spending in the current year is expected to be R35,4 billion, and Gartner says software spending in South Africa will wrow by 12.6 Percent in 2019. The software development project success rate of 77% of this study proves that the projects will be successful.

According to Pretorius et al. (2012), there is no relationship between project management maturity and project outcome because project can be successful despite the maturity level of the organisation. This is contrary to previous studies such as the CHAOS Standish Report and the Prosperus Report, which found the existence of a medium relationship between the two variables. Pretorius et al. (2012) has also found that the critical project factors influence the success of the project. Researchers such as Yazici (2009), Jimenez *et al.* (2012) have found that project management maturity is positively correlated to certain success criteria on the level of the organization and have listed the benefits of higher level of maturity.

The literature has revealed a number of factors that impact the success of software development projects. In this research study, different critical success factors that were different from those reported in other studies were established. This research study is in agreement with Joseph & Marnewick (2014) about two of the top four success factors, although it differs with the results of the longitudinal study carried out by the Standish Group. According to the list of the top 10 ranked software project factors, the '*committed and motivated of the team'* factor is ranked number one and the technical related factor called *'support tools and good infrastructure*' is ranked number ten. There is a huge need to constantly motivate the members of a project team so as to increase the success rate of the project.

In the chapter that follows, the issue of whether the purpose of this work (i.e. to determine whether the level of software project management maturity influences project success rate) is addressed.

CHAPTER 5: FINDINGS, RECOMMENDATIONS AND CONCLUSIONS

A summary of the findings of the main topics covered by the study are presented in this chapter. The chapter concludes by referring to the aim and objectives that were outlined at the beginning of this dissertation. The main aim is to determine whether the level of software project management maturity influences the success rate of a project. Maturity model can be defined as a way of measuring the status of an organisation regarding its ability to manage projects successfully. The quantitative study that was undertaken adopted CMMI, and 17 constructs where used to determine the exact maturity level.

The most popular assumption about the existence of maturity models is if the organisation has achieved a high maturity level, the organisation should be necessarily be very successful when undertaking future projects. This study was academically structured. The quantitative study was conducted within the positivism philosophy. The research strategy employed was a survey questionnaire and SPSS together with MS Excel were used as the statistical analysis programs.

The main goal of the chapter is to address the research findings presented in Chapter 4. This chapter is structured as follows: The first part is an introduction, and section 5.2 provides an overview of all the chapters of the dissertation. Section 5.3 confirms that the research question was answered, and the research goals and objectives were met. The limitations of the research study as well as the recommendations are presented in section 5.4 and 5.5. After outlining the contribution of the study (section 5.6), the chapter concludes with a personal reflection of the study by the researcher.

5.2 OVERVIEW OF CHAPTERS

This section summarises the chapters covered by the dissertation from Chapters 1 to 4. The limitations, findings, recommendations and conclusions are represented on separate sections of Chapter 5. The limitations of the study are recognised before the conclusion section, which informs the reader about what was discovered in the study.

5.2.1 CHAPTER I INTRODUCTION

The opening chapter provided the research problem and research background. The problem statement was formulated thereafter. The research variables were identified, and their definitions were provided.

5.2.2 CHAPTER 2 LITERATURE

Chapter 2 presented the propositions of the study. In this chapter, the researcher demonstrated a deep understanding of all the scholarly work published in this field of study. The literature review revealed that the Standish Group has been monitoring IT, project success rate since 1994 through its CHAOS Reports, while the Prosperus Report was monitoring the success rate in Southern Africa since 2003. The literature review also clarified the current problem and common ideas regarding project success, including the definition of the term ”*success*” by various industries.

5.2.3 CHAPTER 3 METHODOLOGY

All the possible data collection strategies were evaluated and the most appropriate strategy for the study was selected. The justification of the approach used at a philosophical and an operational level was provided. Quantitative methodology was relevant for a number of reasons and was hence selected for this research study since a questionnaire of closed-ended questions was adopted for this research study. Furthermore, the sampling method used was justified in this chapter. Thereafter, the questionnaire was chosen as the most appropriate data collection strategy for descriptive research. The chapter was structured on deductive reasoning.

5.2.4 CHAPTER 4 DATA PRESENTATION

Different types of quantitative methods such as statistics, graphs and tables were used to present the results. The collected data was organised and processed using statistical software. The chapter analysed the survey results and identified the relationships among variables since one variable was found to influence the other variable. The data was reduced to means, standard deviations, correlations and other statistical summarizes. The focus was on average performances instead of individual performances.

The study has highlighted the areas that needs improvement or further research. Since the aim of quantitative research objective is to develop theories and/or hypotheses pertaining to phenomena, the theory was a core discussion of this research study.

5.3 RESEARCH GOALS AND OBJECTIVES

The goal of this research study was to determine whether or not project success rate is influenced by the level of software project management maturity level. Furthermore, the study developed the three major objectives to support the main aim of the study; these will now be discussed.

1. The first objectives was to determine current software project success rates. As illustrated in **Table 4.13**, although software spending in South Africa is increasing every year, the

time taken to complete a software project remains poor when compared with other success dimensions. Another important finding was that the overall success rate established in this research study was 77%, and this differs by 10% when compared with the study of Sauer et al. (2007), which recorded a success rate of 69%. This research study used the project performance metrics employed by Version One Inc. (*VERSIONONE.COM*, 2018). Good performance was noted on all three project performance metrics (i.e. budget, quality and time). The critical success factors which contribute to the success of software development project were identified by this research study and these were illustrated in **Tables 4.5** and **4.7** as well as **Figure 4.6**. The correlation (or rather lack of it) between success factor categories and project outcome was demonstrated in **Table 4.6**; it was established that success factors do not contribute significantly to the software developers project outcome.

2. The second objective was to determine the existence of a relationship between success and maturity. To this end, the relationship between performance outcome and maturity level of a software development project was determined. Whereas it was found that other process areas constructs have a medium relationship with project outcome, no correlation was established between other process areas and project outcome. As illustrated in **Table 4.11**, product integration process was found to contribute to the project outcome, and the verification process was found not to affect the performance of software development project. Relationships between integrated project management and quotation project management (both are process areas of project management) with maturity level were established. This suggests that if maturity level is low, the organisation will also perform badly at quantitative project management and integrated project management levels. Organisation processes performance and organisation performance management showed significance relationship.
3. The third objective was to determine IT project management maturity level, specifically in the area of software development. The study has found improved maturity level as compared to maturity level of IS project found by Marnewick (2013). In this research study, a maturity level of 3.89 was reported in **Figure 4.14**. On the contrary, Marnewick (2013) reported a much lower maturity level of 3.12. The established maturity value of 3.89 for this study was attributed to the fact that the software companies are now comprehensively following the industry standardised processes when the software is under development. Relative to the Marnewick (2013) study, a steady improvement in the maturity level is recorded every five years.

Given the quantitative nature of this research, the researcher relied on numerical data to test the relationship between maturity and project outcome. The post-positivist approach was employed to test the available theories between the two variables, and the answer was found using the above revisited objectives. Based on the results presented in Chapter 4, the research goal was partially achieved. When software development project outcome is defined by three success dimensions, which were categorised into five measures, the project performs much better. The leading measure is the use of delivered product, which shows that the customer is always using the delivered product every time. The second objective, namely the determination of relationship between maturity level and success rate, was also addressed. Being the most popular framework (Jalote, 2002), the CMMI was adopted and used to measure the current level of maturity of software development projects. The CMMI covers organisation and project process issues (Jalote, 2002). A high success rate of 77% and maturity level of 3.89 were observed in the current study. Looking at different process areas that constitute maturity, it was determined that some of the processes have a relationship with project performance.

As illustrated in **Figure 4.13**, the software development organisation performed better on software development process compared to project management process and organisational processes. The statistical significance shows that the relationship between maturity level and software development project performance has a medium relationship. This means that other maturity process areas determine the success of the software development project, and other maturity process areas do not contribute to the success of the project. The success rate was found to increase dramatically by -+30% when compared with the success rate reported in the literature in Chapter 2.

5.4 RESEARCH LIMITATIONS

Similar to other research studies, this study has limitations. Although a large number of respondents were invited, some of the respondents refused to participate because permission was not granted by their companies to participate in this research study. The researcher's desire to travel to other provinces with a view to engage other more experienced members of the software development industry was hampered by lack of funding. Adequate funding would have also allowed more than a single researcher to be engaged in the research work thus allowing enhanced coverage.

The instrument that was used to conduct the research (questionnaire) did not allow for the industry domain of the respondents to be stipulated; software development is viewed as being the same for each organisation. Therefore, the nature of respective organisations of the respondents was not captured and is therefore not known; for example, private sector or public sector organization, those entities were not required by research instrument used.

The primary type of software product provided by the organisations was also not provided; while some organisations provide in-house development (custom built software), others provide ff-the-shelf software and others provide support for off-the-shelf software. The respondents were not allowed to specify their specialisations, so such information was not covered by the survey. Furthermore, no distinction was made between respondents of organisations from the private and public sectors. In addition, information regarding the budget of the projects was excluded because the majority of the software development team members are not familiar with the financial aspects of their projects.

The duration taken to complete the project was also not addressed in great detail. In fact, the definition of the term "*success*" was based on three constructs only, and it was therefore difficult to compare this research study with other longitudinal studies.

It is an assumption that the software development team wants to develop a quality software within a short period of time; this was, however, not part of the study. Furthermore, the complexity of the projects was not addressed in the study. Moreover, the project success was based on the triple constraint, which is a common definition of success.

Most of the projects are medium size projects. Different Software development projects requires different technical support, which was not covered by the questionnaire.

Other project team members were not revealed by the respondents, and the researcher is only familiar with only software developers, project managers, BIs, testers and others who were mentioned in the questionnaire. The list of IT job titles is evolving as the industry grows; some organisations have job titles such as technical operations officer and software architect. Therefore, it is possible that the researcher might have missed potential respondents because they might have acquired new titles, which are not commonly known. It was expected of the respondents to extend the invitation to any members of their software projects team members that might have been missed out by the researcher. The results are based on the knowledge and attitude of the respondents and not be reflective of the organisations they represent.

In all cases, the size of the organisations was not disclosed, only the project size was revealed by the research instrument. Therefore, it is not known if the small organisations are handling big projects or *vice versa*.

The survey questionnaire was designed based on CMMI software development, project management and organisational goals. CMMI integrates multiple models that can be applied in different industries. Each process area has specific goals, and each specific goal has a list of specific practices that need to be satisfied. Upon completion of the evaluation process, the specific maturity level was determined based on the level of specific practice evidence. There is a general shortage of literature on other maturity models. For this reason, the CMMI is regarded as the most popular maturity model for software industry (Jalote, 2002), and was therefore adopted in this research study. According to Seelhofer & Graf (2018), CMMI with its five maturity levels has been applied in the engineering and construction, telecommunication, IT and manufacturing industries. In the section that follows, suggestions on how the research can be taken further is presented.

5.5 RECOMMENDATION FOR FUTURE RESEARCH

The research study has suggested future research because of the nature of the industry. Therefore, the following recommendations are made based on the findings of this research:

Further study is required to investigate why there is such a huge difference in terms of the success rate reported by the Standish Group (30%), the Prosperus Report (60%) and this study (77%). Care should be taken to ensure that the recommended/future study should be longitudinal and thus be aligned with the CHAOS Report and the Prosperus Report to allow the monitoring and reporting of any significant changes on a regular basis. The figures reported by Gartner (2018) show how much is South Africa spending on IT related projects. A question that comes to mind is: does it mean that IT in South Africa are over-priced when compared with other countries?

The factor "*committed and motivated team*" was ranked as the number 1 critical success factor in this research study, and "*clear requirements and specifications*" was ranked as the number 1 critical success factor by Nasir & Sahibuddin (2011) and the Prosperus Report (2014). This demonstrates a need for future research to be undertaken on critical project factors that are dominant in the literature. Other critical success factors that also require further research include "*user involvement"* and *"good leadership*"; very little research has been conducted on these two factors. Two additional factors that require attention are factors with relatively low mean values; these factors are "*end-user training provision*" and "*frozen requirement*", which have received ratings of 3.96 and 3.38, respectively. At 3.99, the "*effective monitoring and control*" factor faired a lot better, but would however still require attention because it is not far off from the "*end-use training provision*" factor.

Software development organisations should assess their maturity levels. Organisations must start with their development team and assess the maturity level, and finally assess the organisation as a whole. Questions that need to be asked include: does the organisation that specialise in project management receive value of project management maturity? or do the companies experience challenges when determining their exact maturity model? The maturity models were promoted back in the 1980s, and has the importance of maturity models increased or decreased since then? Further studies to establish whether or not the leading software organisations have interests in maturity needs also need to be undertaken. There is also a need to determine if the organisation maturity have an effect to its software development team if the team is not maturity oriented.

Future studies might also involve determining all variables that must be considered to form part of the definition of the project success; in the software development project industry, phrases such as "*high employee morale*" are not used regularly as part of a definition of project success. Furthermore, this study has not established whether the size of the software development project has an effect on the success of a project. For example, does the success rate double if the project is medium or large? According to the findings of this research study, the projects have a 77% rate of success, and the majority of projects are medium followed by large.

The research study determined the main critical success factors contributing to the software project success as analysed in table 4.5. The figures presented provide practical significance and guidelines to practitioners and business leaders with regards to software project success. Software development industry experts and business leaders should collaborate and work closer to each other. The industry should not collapse in front of the practitioners, something needs to happen. The practitioners should replace their current maturity model if the benefits are not visible. Software projects organisations must deploy a maturity model that fits in to their business model. On this note, the researcher's recommendation to practitioners and business leaders is to consider Agile or Devops, then as the researcher we will perform the comparative or relational studies to report the performance.

The project manager is one person held responsible when projects fail. However, the success of any project is dependent on the team members that are involved in the execution of that particular project. To this end, project failure risks can be reduced by distributing some responsibilities to other team members. Unidirectional leadership must therefore be discouraged, and the project manager should share the leadership of the project with other senior members of the project team. Literature is replete with examples where senior project managers influence the success of the project. The relationships between a client and project manager must also be maintainable. It is no longer easy for a project manager to have a direct relationship with modern generation of software development team members; therefore, there is a need for other representatives to communicate directly with the team. Given the above discussion, it is therefore important to establish on who is the outcome of the whole project dependents on. The impact of managing many software development projects on project outcome therefore needs to be investigated

5.6 CONTRIBUTIONS OF THE RESEARCH

The study contributes to the academic body of knowledge and to project management community as a whole. Furthermore, the factors that affect the success of software development projects were studied and identified. The theoretical contribution of this study was realised in big volumes in different sections of this study. The research will contribute largely to a discussion on the effect of maturity level on the software industry, which was raised by many researchers such as Albrecht & Spang (2011), Farrokh & Mansur (2013), Lianying, Jing & Xinxin (2012), Marnewick (2013a) and Mittermaier & Steyn, 2009.

5.7 SELF REFLECTION

No single factor can define project success. In fact, project success is defined in terms of schedule, quality and budget.

The study surveyed all the member categories of the software development project team. Survey data was collected from developers, project managers, testers, business analysts and others.

Based on 26 critical success factors established by Nasir & Sahibuddin (2011), the study identified 10 factors that appear to be more important than the rest. All critical success factors that were used in this study have a positive response of 176 respondents. Results of this study indicate that 86% of the respondents view the factor of "*committed and motivated team*" as a key critical factor.

At a practical level, a high maturity level translates into highly successful projects. CMMI was chosen by default based on its international recognition and transparent evaluation process.

Agarwal, N. and Rathod, U. (2006) 'Defining "success" for software projects: An exploratory revelation', *International Journal of Project Management*, 24(4), pp. 358–370. doi: 10.1016/j.ijproman.2005.11.009.

Ahmed, A. (2012) *Software Project Templates*, *Software Project Management*. Boca Raton : CRC Press. doi: 10.1201/b11362-30.

Albrecht, J. . and Spang, K. (2011) 'Towards an understanding of the relationship between project management maturity and project success.', *In International Conference on Current Issues in Management of Business and Society Development*, pp. 14–22.

Andersen, E. S. and Jessen, S. A. (2003) 'Project maturity in organisations', *International Journal of Project Management*, 21(6), pp. 457–461. doi: 10.1016/S0263-7863(02)00088-1.

Andersen, E. S. and Jessen, S. A. (2007) 'A Quick Measure of Project Maturity', in *2007 PMI Global Congress Proceedings*. Available at: https://www.pmi.org/learning/library/measure-project-maturity-knowledge-attitude-actions-7336 (Accessed: 27 May 2019).

Archibald, R. and Prad, D. (2014) 'Introduction to maturity in project management', *pmworldlibrary.net*, 3(1). Available at: https://pmworldlibrary.net/wp-content/uploads/2014/01/pmwj18-jan2014-Archibald-Prado-Introduction-MaturitySeriesArticle1.pdf (Accessed: 27 May 2019).

Arias, G. *et al.* (2012) 'The 7 key factors to get successful results in the IT Development projects', *Procedia Technology*, 5, pp. 199–207. doi: 10.1016/j.protcy.2012.09.022.

Association for Project Management. (2012) *APM body of knowledge.* Available at: http://oasis.unisa.ac.za/search/?searchscope=1&searchtype=i&searcharg=1903494400 (Accessed: 27 May 2019).

Attarzadeh, I. and Ow, S, E. (2008) 'Project Management Practices : Success versus Failure', 1, pp. 2–9.

Backlund, F., Chronéer, D. and Sundqvist, E. (2014) 'Project Management Maturity Models – A Critical Review: A Case Study within Swedish Engineering and Construction Organizations', *Procedia - Social and Behavioral Sciences*, 119, pp. 837–846. doi: http://dx.doi.org/10.1016/j.sbspro.2014.03.094.

Balnaves, M. and Caputi, P. (2001) *Introduction to quantitative research methods : an investigative approach*. London ; SAGE.

Bay, A. and Skitmore, M. (2006) *PROJECT MANAGEMENT MATURITY: SOME RESULTS FROM INDONESIA*, *Journal of Building and Construction Management*. Available at:

http://eprints.qut.edu.au/9413/1/9413.pdf (Accessed: 7 July 2019).

Belassi, W. and Tukel, O. I. (1996) 'A new framework for determining critical success/failure factors in projects', *International Journal of Project Management*, 14(3), pp. 141–151. doi: 10.1016/0263-7863(95)00064-X.

Bergerm, S. and Freund, A. (2012) 'Fear of Failure, Disorganization, and Subjective Well-Being in the Context of Preparing for an Exam', *Article in Swiss Journal of Psychology*, 71(2), pp. 83–91. doi: 10.1024/1421-0185/a000074.

Berndtsson, M. (2008) *Thesis projects : a guide for students in computer science and information systems*. London: Springer. Available at: https://www.worldcat.org/title/thesis-projects-a-guide-for-students-in-computer-science-and-information-systems/oclc/938657616&referer=brief_results (Accessed: 1 June 2019).

Bhattacherjee, A. (2012) *Introduction to Research, Social Science Research: Principles, Methods, and Practices*. Second edi, *University of South Florida*. Second edi. doi: 10.1351/pac198961091657.

Biggam, J. (2015) *Succeeding with your master's dissertation : a step-by-step handbook*. McGraw Hill/Open University Press.

Bourne, L. (no date) *CMMI | Mosaicproject's Blog*. Available at: https://mosaicprojects.wordpress.com/tag/cmmi/ (Accessed: 1 September 2018).

Burke, R. (2011) 'Advanced Project Management: Fusion Method Xyz: a Project Methodology Systems Approach for the Project Sponsor to Implement Corporate Strategy'. Available at: https://scholar.google.co.za/scholar?hl=en&as_sdt=0%2C5&q=Burke+%282011%29%2C+Project+management&btnG= (Accessed: 21 June 2019).

Cameron, S. and Price, D. (2009) *Business Research Methods : a Practical Approach.* London : CIPD. Available at: https://www.worldcat.org/title/business-research-methods-a-practical-approach/oclc/944042158&referer=brief_results (Accessed: 3 May 2019).

Camilleri, E. (2016) *Project success : critical factors and behaviours*. 1st ed. Routledge. Available at: https://www.worldcat.org/title/project-success-critical-factors-and-behaviours/oclc/1100062466&referer=brief_results (Accessed: 1 June 2019).

Cheema, A. and Shahid, A. A. (2005) 'Customizing Project Management Methodology', in *2005 Pakistan Section Multitopic Conference*. IEEE, pp. 1–6. doi: 10.1109/INMIC.2005.334390.

Chemuturi, M. (2011) *Mastering software quality assurance : best practices, tools and techniques for software developers*. Fort Lauderdale Fla.: J. Ross Pub. Available at:

https://www.worldcat.org/title/mastering-software-quality-assurance-best-practices-tools-and-techniques-for-software-developers/oclc/847506143&referer=brief_results (Accessed: 31 May 2019).

Chemuturi, M. (2013) *Requirements engineering and management for software development projects*. New York: Springer. Available at: https://www.worldcat.org/title/requirements-engineering-and-management-for-software-development-projects/oclc/842897871&referer=brief_results (Accessed: 31 May 2019).

Chemuturi, M. and Cagley, T. M. (2010) *Mastering software project management : best practices, tools and techniques*. Fort Lauderdale Fla.: J. Ross Pub. Available at: https://www.worldcat.org/title/mastering-software-project-management-best-practices-tools-and-techniques/oclc/769225698&referer=brief_results (Accessed: 31 May 2019).

Chrissis, M. B. *et al.* (2011) *CMMI for development : guidelines for process integration and product improvement*. Upper Saddle River, NJ : Addison-Wesley. Available at: https://www.worldcat.org/title/cmmi-for-development-guidelines-for-process-integration-and-product-improvement/oclc/718419516&referer=brief_results (Accessed: 1 June 2019).

Chrissis, M. and Weber, C. (1993) *Capability Maturity Model for Software, Version 1.1*. Available at: https://www.worldcat.org/title/capability-maturity-model-for-software-version-11/oclc/831694651&referer=brief_results (Accessed: 1 June 2019).

Clinning, G. (2016) *Information technology project managers' commitment to sustainability*. Available at: https://ujcontent.uj.ac.za/vital/access/services/Download/uj:20923/SOURCE1 (Accessed: 1 June 2019).

Coletta, A. (2007) 'An industrial experience in assessing the capability of non-software processes using ISO/IEC 15504', *Software Process: Improvement and Practice*. John Wiley & Sons, Ltd, 12(4), pp. 315–319. doi: 10.1002/spip.320.

Cooke-davies, T. J. and Arzymanow, A. (2003) 'The maturity of project management in different industries : An investigation into variations between project management models', 21, pp. 471–478. doi: 10.1016/S0263-7863(02)00084-4.

Cooper, P. W. (2013) *Insights and Trends: Current Portfolio, Programme, and Project Management Practices The third global survey on the current state of project management*.

Devlin, A. S. (2006) *Research methods : planning, conducting and presenting research*. Belmont CA [etc.]: Thomson/Wadsworth.

Dolado, J. (2000) 'A validation of the component-based method for software size estimation', *ieeexplore.ieee.org*, 26(10), pp. 1006–1021. Available at: https://ieeexplore.ieee.org/abstract/document/879821/ (Accessed: 1 June 2019).

Donaldson, S. E. and Siegel, S. G. (1997) *Cultivating successful software development : a practitioner's view*. Prentice Hall.

Dorsey, P. (2005) 'Top 10 Reasons Why Systems Projects Fail', pp. 1–9.

Ehsan, N. *et al.* (2010) 'CMMI / SPICE based process improvement', *Management of Innovation and Technology (ICMIT), 2010 IEEE International Conference on*, pp. 859–862. doi: 10.1109/ICMIT.2010.5492803.

Eickelman, N. (2003) 'An insider's view of CMM Level 5', *ieeexplore.ieee.org*, 20(4), pp. 79–81. Available at: https://ieeexplore.ieee.org/abstract/document/1207461/ (Accessed: 12 August 2019).

Erasmus, W. and Marnewick, C. (2012) 'Project management-The saviour in turbulent times? A cross-sector analysis.', in. Available at: https://scholar.google.co.za/scholar?hl=en&as_sdt=0%2C5&q=Erasmus+%26+Marnewick+%282012%29&btnG=#d=gs_cit&u=%2Fscholar%3Fq%3Dinfo%3AKMvm72oEkxIJ%3Ascholar.google.com%2F%26output%3Dcite%26scirp%3D0%26hl%3Den.

Eriksson, P. and Kovalainen, A. (2011) *Qualitative Methods in Business Research*, *SAGE Publications Ltd.* Available at: https://books.google.co.za/books?id=9QJAwErOjM4C&printsec=frontcover&dq=qualitative+methods+in+business+research.+eriksson&hl=en&sa=X&ved=0ahUKEwjd0avzpIDjAhWgThUIHeehAsQQ6AEIKjAA#v=onepage&q=qualitative methods in business research. eriksson&f= (Accessed: 23 June 2019).

Eriksson, P. and Kovalainen, A. (2015) *Qualitative methods in business research: A practical guide to social research.* Reprinted. London: SAGE. Available at: https://www.worldcat.org/title/qualitative-methods-in-business-research/oclc/814092108&referer=brief_results (Accessed: 1 June 2019).

Farrokh, J. and Mansur, A. K. (2013) 'Project Management Maturity Models and Organizational Project Management Maturity Model (OPM3 ®): A Critical Morphological Evaluation', 7(5), pp. 1102–1105.

Fontana, R. M. *et al.* (2014) 'Processes versus people: How should agile software development maturity be defined?', *Journal of Systems and Software*. Elsevier Inc., 97, pp. 140–155. doi: 10.1016/j.jss.2014.07.030.

Galin, D. (2004) *Software Quality Assurance From theory to implementation Software Quality Assurance From theory to implementation CYAN MAGENTA YELLOW BLACK*. Available at: www.pearsoned.co.uk (Accessed: 7 July 2019).

Gartner Says IT Spending in South Africa Will Grow 4.3 Percent in 2018 (no date). Available at: https://www.gartner.com/en/newsroom/press-releases/2018-08-27-gartner-says-it-spending-in-south-africa-will-grow-4-percent-in-2018 (Accessed: 31 May 2019).

Garzás, J. *et al.* (2013) 'A maturity model for the Spanish software industry based on ISO standards', *Computer Standards & Interfaces*, 35(6), pp. 616–628. doi: 10.1016/j.csi.2013.04.002.

Gheni, A., Jusoh, Y. and Jabar, M. (2017) 'The Critical Success Factors (CSFs) for IT projects', *journal.utem.edu.my*, 9(3–3), pp. 13–17. Available at: http://journal.utem.edu.my/index.php/jtec/article/view/2865 (Accessed: 1 June 2019).

Goddard, W. and Melville, S. (2001) *Research methodology : an introduction*. 2nd ed. Kenwyn South Africa: Juta. Available at: https://www.worldcat.org/title/research-methodology/oclc/953628816&referer=brief_results (Accessed: 2 June 2019).

Gomes, J., Romão, M. and Carvalho, H. (2016) 'Successful IS/IT projects in Healthcare: Pretesting a questionnaire', *Elsevier*, 100, pp. 375–382. Available at: https://www.sciencedirect.com/science/article/pii/S1877050916323407 (Accessed: 1 June 2019).

Gordon, L. (2019) *Real research: Research methods sociology students can use*. Second edi. Available at: https://www.worldcat.org/title/real-research-research-methods-sociology-students-can-use/oclc/1055570026&referer=brief_results (Accessed: 1 June 2019).

Greener, S. (2008) *Business research methods*. Available at: https://www.worldcat.org/title/business-research-methods/oclc/893605820&referer=brief_results (Accessed: 1 June 2019).

Greener, S. and Thomas, M. J. (2015) *Introduction to Business Research Methods*.

Grobler, P. and Steyn, H. (2006) 'Project Management Maturity Models: Does One Size Fit All', *researchgate.net*, pp. 149–157. Available at: https://www.researchgate.net/profile/H_Steyn2/publication/287760085_Project_management_maturity_models_Does_one_size_fit_all/links/57d9b43c08ae601b39b16014/Project-management-maturity-models-Does-one-size-fit-all.pdf (Accessed: 1 June 2019).

Grottke, M. (2002) 'Software Process Maturity Model Study', *University of Erlangen-Nuremberg, IST-1999-55017*.

Hairul, M., Nasir, N. and Sahibuddin, S. (2011) 'Critical success factors for software projects : A comparative study', 6(10), pp. 2174–2186. doi: 10.5897/SRE10.1171.

Håkansson, A. (2013) 'Portal of research methods and methodologies for research projects and degree projects', *diva-portal.org*, pp. 67–73. Available at: http://www.diva-portal.org/smash/record.jsf?pid=diva2:677684 (Accessed: 1 June 2019).

Hastie, S. and Wojewoda, S. (2015) *Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch*. Available at: https://www.infoq.com/articles/standish-chaos-2015/ (Accessed: 1 June 2017).

Hastings, T. and Sajeev, A. (2001) 'A vector-based approach to software size measurement and effort estimation', *ieeexplore.ieee.org*, 27(4), pp. 337–350. Available at: https://ieeexplore.ieee.org/abstract/document/917523/ (Accessed: 1 June 2019).

Hofstee, E. (2006) *Constructing a good dissertation : a practical guide to finishing a Master's, MBA or PhD on schedule*. Available at: https://www.worldcat.org/title/constructing-a-good-dissertation-a-practical-guide-to-finishing-a-masters-mba-or-phd-on-schedule/oclc/70875130&referer=brief_results (Accessed: 1 June 2019).

Hollister, C. (2014) *Sources: Handbook of Academic Writing for Librarians*. Revised ed, *Reference & User Services Quarterly*. Revised ed. doi: 10.5860/rusq.53n3.269b.

Hughes, B., Ireland, R. and West, B. (2004) *Project Management for IT-related Projects: Textbook for the ISEB Foundation Certificate in IS Project Management*. Available at: https://books.google.co.za/books?hl=en&lr=&id=d0zdfbudN-0C&oi=fnd&pg=PR1&dq=Project+Management+for+IT-related+Projects:+Textbook+for+the+ISEB+Foundation+Certificate+in+IS+Project+Management&ots=6Pf6L-9lJ7&sig=nxKs2RSooS9Wu7z-Swcj88tmfx0 (Accessed: 1 June 2019).

Humble, J. and Russell, R. (2009) 'The agile maturity model applied to building and releasing software.', *THoughtWorks White paper*.

Humphrey, W. S. (1997) *Comments on Software Quality*. Available at: https://pdfs.semanticscholar.org/b5fa/ed27fec2c33c84c3e9a78ab99d71ccfffe30.pdf (Accessed: 7 July 2019).

Hwang, S. M. (2009) 'CMMI and K-model. International Journal of Software Engineering and Its Applications', *International Journal of Software engineering and Its applications*, 3(1), pp. 33–42. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.432.7423&rep=rep1&type=pdf (Accessed: 1 June 2019).

Ibbs, C. W., Reginato, J. M. and Kwak, Y. H. (2004) 'Developing Project Management Capability: Benchmarking, Maturity, Modeling, Gap Analyses, and ROI Studies', in *The Wiley Guide to Managing Projects*. Hoboken, NJ, USA: John Wiley & Sons, Inc., pp. 1214–1233. doi: 10.1002/9780470172391.ch48.

Ibbs, W. C. and Reginato, J. (2002) 'Can good project management actually cost less.', in, pp. 3–10.

Jacobs, S. (2015) *Engineering information security: The application of systems engineering concepts to achieve information assurance*. Available at: https://books.google.co.za/books?hl=en&lr=&id=FeZbCwAAQBAJ&oi=fnd&pg=PR23&dq=Engineering+information+security:+the+application+of+systems+engineering+concepts+to+achieve+information+assurance&ots=p38XtMz0aU&sig=oArZWVGLA80vOI6hs3tUHK1GQqU (Accessed: 1 June 2019).

Jakobsen, C. R. and Johnson, K. A. (2008) 'Mature agile with a twist of CMMI', *Proceedings - Agile 2008 Conference*, pp. 212–217. doi: 10.1109/Agile.2008.10.

Jalote, P. (2002) *Software project management in practice*. Boston : Addison-Wesley. Available at: https://www.worldcat.org/title/software-project-management-in-practice/oclc/850110497&referer=brief_results (Accessed: 1 June 2019).

James, E. A. (2015) *Writing your doctoral dissertation or thesis faster : a proven map to success*. Los Angeles: SAGE. Available at: https://www.worldcat.org/title/writing-your-doctoral-dissertation-or-thesis-faster-a-proven-map-to-success/oclc/1039240843&referer=brief_results (Accessed: 1 June 2019).

Jiang, J. J. . J. J. *et al.* (2004) 'An exploration of the relationship between software development process maturity and project performance', *Information and Management*, 41(3), pp. 279–288. doi: 10.1016/S0378-7206(03)00052-1.

Jimenez, D., Martinez, C. and Martinez, L. (2012) 'EFQM process criteria results. A project management maturity perspective', in.

Jones, C. (2010) *Software engineering best practices : lessons from successful projects in the top companies*. New York : McGraw-Hill. Available at: https://www.worldcat.org/title/software-engineering-best-practices-lessons-from-successful-projects-in-the-top-companies/oclc/777151518&referer=brief_results (Accessed: 1 June 2019).

Jones, C. (2013) 'Function points as a universal software metric', *ACM SIGSOFT Software Engineering Notes*, 38(4), p. 1. doi: 10.1145/2492248.2492268.

Jones, C. (2015) *Software Risk Master™ (SRM) Sizing and Estimating Examples*. Available at: http://namcookanalytics.com (Accessed: 2 June 2019).

Joseph, N. (2014) *A predictive model for information technology project success*. Available at: https://ujcontent.uj.ac.za/vital/access/services/Download/uj:10916/CONTENT1 (Accessed: 1 June 2019).

Joseph, N. and Marnewick, C. (2014) 'Structured Equation Modeling for Determing ICT Project Success Factors'. Available at: https://www.pmi.org/learning/library/structured-equation-modeling-ict-project-success-10616 (Accessed: 1 June 2019).

Judgev, K. and Thomas, J. (2002) ‘Project management maturity models: The silver bullets of competitive advantage’, *Project Management Journal*, 33. Available at: https://prism.ucalgary.ca/bitstream/handle/1880/44250/2002 PMJ PM maturity models.pdf?sequence=1 (Accessed: 1 June 2019).

Kabir, M. and Rusu, L. (2016) ‘IT project development using capability maturity model’, *igi-global.com*, 29(4), pp. 35–48. Available at: https://www.igi-global.com/article/it-project-development-using-capability-maturity-model/164898 (Accessed: 18 June 2020).

Kahate, A. (2004) *Object Oriented Analysis & Design*. Available at: https://books.google.co.za/books?hl=en&lr=&id=PkVy_zAaruMC&oi=fnd&pg=PP19&dq=Object+Oriented+Analysis+%26+Design,+kahate&ots=8Qvea1HKHE&sig=PyfY7QoJJtkVd4O1xBCIT4gdZjc (Accessed: 1 June 2019).

Kaur, B. and Aggrawal, H. (2013) ‘Critical failure factors in information system: an exploratory review’, *pdfs.semanticscholar.org*, 4(1), pp. 76–82. Available at: https://pdfs.semanticscholar.org/8376/1d32613e38467fee0cd4c1a1a329143859c1.pdf (Accessed: 20 February 2019).

Kaur, J. (2014) ‘Comparative Study of Capability Maturity Model’, *International Journal of Advanced research*, 2(1), pp. 47–49. Available at: http://www.ijarcst.com/doc/vol2-issue1/jasleen.pdf (Accessed: 31 May 2019).

Kaur, R. and Sengupta, J. (2013) ‘Software Process Models and Analysis on Failure of Software Development Projects’. Available at: http://arxiv.org/abs/1306.1068 (Accessed: 31 May 2019).

Kelsey, R. (2006) ‘Software Project Management: Measures for Improving Performance’.

Kenett, R. and Baker, E. (2010) *Process Improvement and CMMI® for Systems and Software*. Available at: https://content.taylorfrancis.com/books/download?dac=C2009-0-11626-6&isbn=9781420060515&format=googlePreviewPdf (Accessed: 1 June 2019).

Kerzner, H. (2013) *Project management workbook to accompany Project management - a systems approach to planning, scheduling and controlling, eighth edition*. 8th ed. New Jersey: Wiley. Available at: https://www.worldcat.org/title/project-management-workbook-to-accompany-project-management-a-systems-approach-to-planning-scheduling-and-controlling/oclc/644958458&referer=brief_results (Accessed: 1 June 2019).

Keuten, T. and MacFadyen, T. (2007) *Collaborative Opportunities for Using OPM3 ® and CMMI ®*. Available at: https://www.mosaicprojects.com.au/PDF/OPM3_CMMI_Collaborative_Opportunities.pdf (Accessed: 1 June 2019).

Khanzode, V. (2011) *Research methodology : techniques and trends*. New Delhi: APH Pub. Available at: https://www.worldcat.org/title/research-methodology-techniques-and-trends/oclc/54810771&referer=brief_results (Accessed: 1 June 2019).

Khoshgoftar, M. and Osman, O. (2009) 'Comparison of maturity models', in *2009 2nd IEEE International Conference on Computer Science and Information Technology*. IEEE, pp. 297–301. doi: 10.1109/ICCSIT.2009.5234402.

Klaus-Rosińska, A. and Kuchta, D. (2017) 'Maturity of project management in construction companies with the use of fuzzy numbers', *AIP Conference Proceedings*, 1863. doi: 10.1063/1.4992388.

Kothari, C. (2004) *Research methodology: Methods and techniques*. Available at: https://books.google.co.za/books?hl=en&lr=&id=hZ9wSHysQDYC&oi=fnd&pg=PA2&dq=Research+methodology:+Methods+and+techniques.+&ots=1sYgqJiZB9&sig=xo6nYTj6E258Sgns-QmdzmK-dxs (Accessed: 1 June 2019).

Kulpa, M. K. and Johnson, K. A. (2008) *Interpreting the CMMI : a process improvement approach*. CRC Press. Available at: https://www.worldcat.org/title/interpreting-the-cmmi-r-a-process-improvement-approach-second-edition/oclc/1058331307&referer=brief_results (Accessed: 1 June 2019).

Kumar, R. (2011) *Research methodology: A step-by-step guide for beginners.* Sage Publications Limited.

Kwak, Y. H. and Ibbs, C. W. (2002) 'Project Management Process Maturity (PM)2 Model', *Journal of Management in Engineering*, 18(3), pp. 150–155. doi: 10.1061/(ASCE)0742-597X(2002)18:3(150).

Labuschagne, L., Jakovljevic, M. and Marnewick, C. (2009) *The Prosperus Report 2008: IT Project Management Maturity vs. Project Success in South Africa*. Available at: https://remotevoicetechnologies.com/publications/Publications_2000_to_2010/26_Prosperous_Report_2009_Labuschagne_Marnewick_Jakovljevic_February_First_draft.pdf (Accessed: 1 June 2019).

Larrucea, X. *et al.* (2016) 'Software process improvement in very small organizations', *IEEE Software*. doi: 10.1109/MS.2016.37.

Leedy, P. D. and Ormrod, J. E. (2010) *Practical research : planning and design*. 10th edn. Harlow, Essex : Pearson. Available at: https://www.worldcat.org/title/practical-research-planning-and-design/oclc/859837317&referer=brief_results (Accessed: 1 June 2019).

Leedy, P. and Ormrod, J. (2010) *Practical Research planning and design*. 9th edn. Boston.

Lehtinen, T. *et al.* (2014) 'Perceived causes of software project failures–An analysis of their relationships', *Elsevier*, 56(6), pp. 623–643. Available at:

https://www.sciencedirect.com/science/article/pii/S0950584914000263 (Accessed: 1 June 2019).

Levine, D. M. (2010) *Business statistics : a first course (Computer file, 2010) [WorldCat.org]*. Pearson Education India. Available at: https://www.worldcat.org/title/business-statistics-a-first-course/oclc/779115893&referer=brief_results (Accessed: 1 June 2019).

Lianying, Z., Jing, H. and Xinxin, Z. (2012) 'The project management maturity model and application based on PRINCE2', *Elsevier*, pp. 3691–3697. Available at: https://www.sciencedirect.com/science/article/pii/S1877705812005644 (Accessed: 1 June 2019).

Lipschutz, S. and Schiller, J. J. (2012) *Introduction to probability and statistics*. New York: McGraw-Hill. Available at: https://www.worldcat.org/title/schaums-outlines-introduction-to-probability-and-statistics/oclc/979659514&referer=brief_results (Accessed: 1 June 2019).

Majeed, N. *et al.* (2013) 'Performance Evaluation of IT Project Management in Developing Countries', *academia.edu*, 04, pp. 68–74. doi: DOI: 10.5815/ijitcs.2013.04.08.

Mani, M., Lyons, K. and Sriram, R. (2010) 'Developing a sustainability manufacturing maturity model', *ws680.nist.gov*, pp. 311–321. Available at: http://ws680.nist.gov/publication/get_pdf.cfm?pub_id=906033 (Accessed: 1 June 2019).

Marchewka, J. (2016) *Information technology project management: Providing measurable organizational value*. Available at: https://books.google.co.za/books?hl=en&lr=&id=rsLlBQAAQBAJ&oi=fnd&pg=PR13&dq=jack+marchewka&ots=9lxDflp0lf&sig=a6bb1-rHMhwDu5UR5XS4bfjwaYI (Accessed: 28 June 2019).

Marchewka, J. T. (2013) *Information technology project management*. 4th edn. John Wiley & Sons.

Maree, K. (2016) *First steps in research*. Available at: https://www.worldcat.org/title/first-steps-in-research/oclc/952206329&referer=brief_results (Accessed: 1 June 2019).

Marnewick, C. *et al.* (2013) *Prosperus Report , The African Edition*. Edited by C. Marnewick.

Marnewick, C. (2013) 'To Mature or not to Mature: The Information Systems Conundrum', *South African Computer Journal*. South African Institute of Computer Scientists and Information Technologists, 51(1), pp. 10–21. doi: 10.18489/sacj.v51i0.186.

McConnell, S. (1993) *From Anarchy to Optimizing - Steve McConnell*. Available at: https://stevemcconnell.com/articles/from-anarchy-to-optimizing/ (Accessed: 7 July 2019).

McLeod, L. and MacDonell, S. G. (2011) 'Factors that affect software systems development project outcomes: A survey of research', *dl.acm.org*, 43(4). Available at: https://dl.acm.org/citation.cfm?id=1978803 (Accessed: 1 June 2019).

McManu, J. and Wood-Harper, T. (2007) 'Understanding the sources of information systems project failure'. Available at: https://www.researchgate.net/profile/John_Mcmanus21/publication/329539985_A_study_in_project_failure/links/5c1138aea6fdcc494fef184b/A-study-in-project-failure.pdf (Accessed: 1 June 2019).

Melville, S. and Goddard, W. (Wayne D. (1996) *Research methodology : [an introduction for science & engineering students]*. Kenwyn South Africa: Juta & Co. Ltd. Available at: https://www.worldcat.org/title/research-methodology-an-introduction-for-science-engineering-students/oclc/37685450&referer=brief_results (Accessed: 1 June 2019).

Mittermaier, H. and Steyn, H. (2009) 'Project management maturity: An assessment of maturity for developing pilot plants', *scielo.org.za*, 20(1), pp. 95–108. Available at: http://www.scielo.org.za/scielo.php?script=sci_arttext&pid=S2224-78902009000100009 (Accessed: 1 June 2019).

Mnkandla, E. and Marnewick, C. (2011) 'Project management training : the root cause of project failures?', *Journal of Contemporary Management*. Sabinet, 8(1), pp. 76–94. Available at: https://www.ingentaconnect.com/content/sabinet/jcman/2011/00000008/00000001/art00004 (Accessed: 1 June 2019).

Morgan, D. L. (2017) *Integrating qualitative and quantitative methods : a pragmatic approach*. SAGE Publications Inc. Available at: https://www.worldcat.org/title/integrating-qualitative-and-quantitative-methods-a-pragmatic-approach/oclc/1097113029&referer=brief_results (Accessed: 1 June 2019).

Mtsweni, E. S., Horne, T. and Van der Poll, J. A. (2016) 'Soft Skills for Software Team Members', *International Journal of Computer Theory and Engineering*, 8(2). doi: 10.7763/IJCTE.2016.V8.1035.

Mutafelija, B. and Stromberg, H. (2003) *Systematic process improvement using to ISO 9001:2000 and CMMI*. Boston MA.: Artech House. Available at: https://www.worldcat.org/title/systematic-process-improvement-using-to-iso-90012000-and-cmmi/oclc/850961879&referer=brief_results (Accessed: 1 June 2019).

Mutafelija, B. and Stromberg, H. (2008) *Process Improvement with CMMI v1.2 and ISO Standards*. Auerbach Publications. doi: 10.1201/9781420052848.

Naoum, S. G. (Shamil G. . (2013) *Dissertation research & writing for construction students*. London ;;New York : Routledge. Available at:

https://www.worldcat.org/title/dissertation-research-writing-for-construction-students/oclc/823726457&referer=brief_results (Accessed: 1 June 2019).

Nasir, M. and Sahibuddin, S. (2011) 'Critical success factors for software projects: A comparative study', *researchgate.net*, 6(10), pp. 2174–2186. Available at: https://www.researchgate.net/profile/Mohd_Md_Nasir/publication/228975699_Critical_success_factors_for_software_projects_A_comparative_study/links/540d91120cf2df04e754bd4f/Critical-success-factors-for-software-projects-A-comparative-study.pdf (Accessed: 1 June 2018).

Nazar, S. and Abbasi, E. (2008) 'CMMI and OPM3: Are They Compatible?', *Communications in Computer and Information Science*, pp. 235–242. doi: 10.1007/978-3-540-89853-5_25.

Nenni, M. E. *et al.* (2014) 'How to increase the value of the project management maturity model as a business-oriented framework', *International Journal of Engineering Business Management*, 6. doi: 10.5772/58292.

Neverauskas, B. and Railaite, R. (2013) 'Formation approach for project management maturity measurement', *ecoman.ktu.lt*, 18(2), pp. 360–365. Available at: http://www.ecoman.ktu.lt/index.php/Ekv/article/view/4604 (Accessed: 1 June 2019).

Niazi, M., Wilson, D. and Zowghi, D. (2005) 'A maturity model for the implementation of software process improvement: an empirical study', *Elsevier*, 74(2), pp. 155–172. Available at: https://www.sciencedirect.com/science/article/pii/S0164121203002711 (Accessed: 1 June 2019).

Oates, B. (2005) *Researching information systems and computing*. Available at: https://books.google.co.za/books?hl=en&lr=&id=VyYmkaTtRKcC&oi=fnd&pg=PR11&dq=Researching+information+systems+and+computing&ots=w9WFEWTIYI&sig=wLvD2m2cxd7hPUPeQS6H-OMHC24 (Accessed: 2 May 2019).

Ofori, D. and Deffor, E. W. (2013) 'Assessing Project Management Maturity in Africa: A Ghanaian Perspective', *International Journal of Business Administration*, 4(6), pp. 41–61. doi: 10.5430/ijba.v4n6p41.

Olson, D. L. (2004) *Introduction to information systems project management*. 2nd ed. Maidenhead: McGraw-Hill. Available at: https://www.worldcat.org/title/introduction-to-project-management-a-systems-approach/oclc/51966404&referer=brief_results (Accessed: 2 June 2019).

Olson, D. L. (2014) *Information systems project management*. New York : Business Expert Press,. Available at: https://www.worldcat.org/title/information-systems-project-management/oclc/1041928610&referer=brief_results (Accessed: 1 June 2019).

Olson, D. L. (2015) *Information systems project management*. First edit. Available at: https://www.worldcat.org/title/information-systems-project-management/oclc/902836166&referer=brief_results (Accessed: 2 May 2019).

Pallant, J. (2007) *SPSS survival manual : a step by step guide to data analysis using SPSS for Windows*. 3rd ed. Available at: https://www.worldcat.org/title/spss-survival-manual-a-step-by-step-guide-to-data-analysis-using-spss-version-15/oclc/982723184&referer=brief_results (Accessed: 30 May 2019).

Patel, C. and Ramachandran, M. (2009) 'Agile maturity model (AMM): A Software Process Improvement framework for agile software development practices', *Int. J. of Software Engineering, IJSE*, 2(I), pp. 3–28. doi: 10.4304/jsw.4.5.422-435.

Paulk, M. *et al.* (2003) *Capability maturity model, version 1.1. Technical Report*, *ieeexplore.ieee.org*. Available at: https://ieeexplore.ieee.org/abstract/document/219617/ (Accessed: 1 June 2019).

Paulk, M. C. (1993) 'Comparing ISO 9001 and the Capability Maturity Model for Software', *Software Quality Journal*, 2(4), pp. 245–256. doi: 10.1007/BF00403767.

Paulk, M. C. (1995) *The capability maturity model : guidelines for improving the software process*. Reading, Mass. : Addison-Wesley Pub. Co. Available at: https://www.worldcat.org/title/capability-maturity-model-guidelines-for-improving-the-software-process/oclc/603838704&referer=brief_results (Accessed: 1 June 2019).

Peldzius, S and Ragaisis, S. (2011) 'Comparison of maturity levels in CMMI-DEV and ISO/IEC 15504', *pdfs.semanticscholar.org*, pp. 117–122. Available at: https://pdfs.semanticscholar.org/fcce/916d3b087692a4785813a7df9e8f9c2d970f.pdf (Accessed: 1 June 2019).

Peldzius, Stasys and Ragaisis, S. (2011) 'Investigation correspondence between CMMI-DEV and ISO/IEC 15504', *International Journal of Education and Information Technologies*, 5(4), pp. 361–368.

Pennypacker, J. S. and Grant, K. P. (2003) 'Project Management Maturity: An Industry Benchmark', *Project Management Journal*, 34(1), pp. 4–11. doi: 10.1177/875697280303400102.

Persse, J. R. (2007) *Project management success with CMMI : seven CMMI process areas*. Upper Saddle River [NJ]: Prentice Hall. Available at: https://www.worldcat.org/title/project-management-success-with-cmmi-seven-cmmi-process-areas/oclc/875730162&referer=brief_results (Accessed: 1 June 2019).

Phillips, J. (2004) *IT project management : on track from start to finish*. Emeryville, CA : McGraw-Hill Osborne. Available at: https://www.worldcat.org/title/it-project-management-on-track-from-start-to-finish/oclc/960381670&referer=brief_results (Accessed: 1 June 2019).

Pino, F. J. *et al.* (2010) 'Harmonizing maturity levels from CMMI-DEV and ISO/IEC 15504', *Journal of Software Maintenance and Evolution*, 22(4), pp. 279–296. doi: 10.1002/spip.437.

'PMI ' s Pulse of the Profession TM The High Cost of Low Performance Pulse Perspective' (no date).

Poranen, T. (2010) 'No Title'.

Poranen, T. (2014) 'Software Project Management Summaries 2014'. Available at: https://tampub.uta.fi/bitstream/handle/10024/96057/software_project_management_2014.pdf?sequence=1 (Accessed: 31 May 2019).

Prabhakar, G. (2008) 'What is project success: a literature review', *eprints.uwe.ac.uk*, 3(9), pp. 3–10. Available at: http://eprints.uwe.ac.uk/14460 (Accessed: 1 June 2017).

Pretorius, S., Steyn, H. and Jordaan, J. (2012) 'Project Management Maturity and Project Management Success in the Engineering and Construction Industries in Southern Africa', *The South African Journal of Industrial Engineering*, 23(3), pp. 1–12. doi: 10.7166/23-3-507.

Proença, D. and Borbinha, J. (2016) 'Maturity models for information systems-A state of the art', *Elsevier 'Procedia Computer Science'*, 100, pp. 1042–1049. Available at: https://www.sciencedirect.com/science/article/pii/S1877050916324486 (Accessed: 1 June 2019).

Project Management Institute (2013) *A guide to the project management body of knowledge : (PMBOK® guide)*. PMI. Available at: https://www.worldcat.org/title/guide-to-the-project-management-body-of-knowledge-pmbok-guide/oclc/946536639&referer=brief_results (Accessed: 1 June 2019).

'Project Management Maturity & Value Benchmark 2014' (2014). Available at: https://www.pmsolutions.com/articles/PM_Maturity_2014_Research_Report_FINAL.pdf.

Qassim, A. A. (2008) 'Why information systems projects fail: Guidelines for Successful Projects', *INTOSAI IT J*, pp. 12–17.

Radujković, M. and Sjekavica, M. (2017) 'Project Management Success Factors', in *Procedia Engineering*. Elsevier Ltd, pp. 607–615. doi: 10.1016/j.proeng.2017.08.048.

Remenyi, D. and Academic Conferences International (2014) *Dictionary of research concepts and issues : 600+ essential concepts and issues required for successful academic research*. [Sonning Common UK]: Academic Conferences & Publ. International.

Available at: https://www.worldcat.org/title/dictionary-of-research-concepts-and-issues/oclc/913047034&referer=brief_results (Accessed: 1 June 2019).

Roscoe, J. (1975) *Fundamental research statistics for the behavioral sciences*. 2. ed. New York: Holt Rinehart and Winston. Available at: https://www.worldcat.org/title/fundamental-research-statistics-for-the-behavioral-sciences/oclc/479651742&referer=brief_results (Accessed: 1 June 2019).

Ruhe, G., Wohlin, C. and Springer-Verlag GmbH (2016) *Software Project Management in a Changing World*. Available at: https://www.worldcat.org/title/software-project-management-in-a-changing-world/oclc/1062246654&referer=brief_results (Accessed: 1 June 2019).

Sanchez, O. P., Terlizzi, M. A. and de Moraes, H. R. de O. C. (2017) 'Cost and time project management success factors for information systems development projects', *International Journal of Project Management*. Elsevier Ltd, 35(8), pp. 1608–1626. doi: 10.1016/j.ijproman.2017.09.007.

Sauer, C., Gemino, A. and Reich, B. (2007) 'The impact of size and volatility on IT project performance', *Communications of the ACM*, 50(11), pp. 79–84. Available at: https://www.researchgate.net/profile/Blaize_Reich/publication/220419735_The_impact_of_size_and_volatility_-_On_IT_project_performance/links/0a85e53308694440ac000000.pdf (Accessed: 30 May 2019).

Saunders, M., Lewis, P. and Thornhill, A. (2012) *Research methods for business students*. 6th edn. Prentice Hall. Available at: https://www.worldcat.org/title/research-methods-for-business-students/oclc/669693976&referer=brief_results (Accessed: 31 May 2019).

Schulmeyer, G. G. (2008) *Handbook of software quality assurance*. 4th ed. Boston: Artech House. Available at: https://www.worldcat.org/title/handbook-of-software-quality-assurance/oclc/300159569&referer=brief_results (Accessed: 31 May 2019).

Seelhofer, D. and Olivier Graf, C. (2018) 'National Project Management Maturity: A Conceptual Framework', *Central European Business Review*. University of Economics, 7(2), pp. 1–20. doi: 10.18267/j.cebr.196.

Sekaran, U. (1992) *Research methods for business : a skill-building approach*. 2nd. New York: Wiley.

Sekaran, U. and Bougie, R. (2013) *Research methods for business : a skill-building approach*. 6th ed. Chichester West Sussex: Wiley.

Sekaran, U. and Bougie, R. (2016) *Research methods for business : a skill-building approach*. John Wiley & Sons. Available at: https://www.worldcat.org/title/research-methods-for-business-a-skill-building-approach-seventh-edition/oclc/943211712&referer=brief_results (Accessed: 31 May 2019).

Shelton, C. (2008) *Scrum Alliance Member-Submitted Informational Articles*. Available at: https://www.scrumalliance.org/community/articles/2008/july/agile-and-cmmi-better-together (Accessed: 31 May 2019).

Silva, F. S. *et al.* (2015) *Using CMMI together with agile software development: A systematic review.*, *Information and Software Technology*. doi: 10.1016/j.infsof.2014.09.012.

Silvius, G. *et al.* (2017) *Sustainability in Project Management*.

Sommerville, I. (2011) *Software engineering*. Boston ;;Munich: Pearson. Available at: https://www.worldcat.org/title/software-engineering/oclc/699648396&referer=brief_results (Accessed: 31 May 2019).

Sonnekus; Rudi; Labuschagne, L. (2003) *IT project management maturity versus project success in South Africa. The Prosperus Report*.

De Souza, T. F. and Gomes, C. F. S. (2015) 'Assessment of maturity in project management: A bibliometric study of main models', *Procedia Computer Science*. Elsevier Masson SAS, 55(Itqm), pp. 92–101. doi: 10.1016/j.procs.2015.07.012.

Spalek, S. (2013) 'Influence of Project Management Maturity on Projects' Costs', *toknowpress.net*, pp. S2-79. Available at: http://www.toknowpress.net/ISBN/978-961-6914-07-9/papers/S2_79-85.pdf (Accessed: 31 May 2019).

Subramanian, G. H., Jiang, J. J. and Klein, G. (2007) 'Software quality and IS project performance improvements from software development process maturity and IS implementation strategies', *Journal of Systems and Software*, 80(4), pp. 616–627. doi: 10.1016/j.jss.2006.06.014.

Sukhoo, A. *et al.* (2007) 'An Evolutionary Software Project Management Maturity Model for Mauritius', *Interdisciplinary Journal of Information, Knowledge, and Management*, 2, pp. 099–118. doi: 10.28945/103.

Sutherland, J., Jakobsen, C. R. and Johnson, K. (2008) 'Scrum and CMMI Level 5: The Magic Potion for Code Warriors', in *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*. IEEE, pp. 466–466. doi: 10.1109/HICSS.2008.384.

Tahri, H. and Drissi-Kaitouni, O. (2015) 'New design for calculating Project Management Maturity (PMM)', *Procedia - Social and Behavioral Sciences*, 181, pp. 171–177. doi: 10.1016/j.sbspro.2015.04.878.

Tamura, S. (2009) 'Integrating CMMI and TSP / PSP : Using TSP Data to Create Process Performance Models', (November).

Team, C. P. (2010) *CMMI for development, Version 1.3 : Improving processes for developing better products and services*. Hanscom AFB MA: SEI. Available at: https://www.worldcat.org/title/cmmi-for-development-version-13-improving-processes-for-developing-better-products-and-services/oclc/929681140&referer=brief_results (Accessed: 1 June 2019).

Tsui, F. F., Karam, O. and Bernal, B. (2014) *Essentials of software engineering, third edition*. Third edition /. Jones & Bartlett Learning. Available at: https://www.worldcat.org/title/essentials-of-software-engineering/oclc/824726169&referer=brief_results (Accessed: 31 May 2019).

VERSIONONE.COM (no date). Available at: https://versionone.com/pdf/VersionOne-10th-Annual-State-of-Agile-Report.pdf (Accessed: 31 May 2019).

Walliman, N. (2005) *Your Research Project : a Step-by-Step Guide for the First-Time Researcher.* London ;;Thousand Oaks, Calif. : Sage Publications. Available at: https://www.worldcat.org/title/your-research-project-a-step-by-step-guide-for-the-first-time-researcher/oclc/988946595&referer=brief_results (Accessed: 31 May 2019).

Walliman, N. (2018) *Research methods: The basics*. Available at: https://www.taylorfrancis.com/books/9781315529011 (Accessed: 20 April 2019).

von Wangenheim, C. G. *et al.* (2010) 'Best practice fusion of CMMI-DEV v1. 2 (PP, PMC, SAM) and PMBOK 2008', *Elsevier*, 52(7), pp. 749–757. Available at: https://www.sciencedirect.com/science/article/pii/S0950584910000480 (Accessed: 31 May 2019).

Von Wangenheim, C. G. *et al.* (2010) 'Systematic literature review of software process capability/maturity models', *researchgate.net*. Available at: https://www.researchgate.net/profile/Jean_Hauck/publication/228738042_Systematic_Literature_Review_of_Software_Process_CapabilityMaturity_Models/links/5542922e0cf24107d39472e5/Systematic-Literature-Review-of-Software-Process-Capability-Maturity-Models.pdf (Accessed: 31 May 2019).

Warrilow, S. (2009) *Project Management Maturity Model*. Available at: https://www.projectsmart.co.uk/project-management-maturity-model.php (Accessed: 12 March 2018).

Wateridge, J.-I. journal of project (1998) 'How can IS/IT projects be measured for success?', *Elsevier*, 16(1), pp. 59–63. Available at: https://www.sciencedirect.com/science/article/pii/S0263786397000227 (Accessed: 31 May 2019).

Wegner, T. (2012) *Applied business statistics : methods and Excel-based applications*.

Claremont, South Africa : Juta,. Available at: https://www.worldcat.org/title/applied-business-statistics-methods-and-excel-based-applications/oclc/867773353&referer=brief_results (Accessed: 31 May 2019).

Welman, J. C. and Kruger, F. (2001) *Research methodology for the business and administrative sciences*. Oxford University Press Southern Africa.

Woodwell, D. (2014) *Research Foundations: How Do We Know What We Know?* 1 Oliver's Yard, 55 City Road London EC1Y 1SP : SAGE Publications, Ltd. doi: 10.4135/9781506374734.

Yazici, H. J. (2009) 'The Role of Project Management Maturity and Organizational Culture in Perceived Performance', *Project Management Journal*. SAGE PublicationsSage CA: Los Angeles, CA, 40(3), pp. 14–33. doi: 10.1002/pmj.20121.

Yeo, K. T. (2002) 'Critical failure factors in information systems projects', *International Journal of Project Management*, 20, pp. 241–246. doi: 10.1016/S0263-7863(01)00075-8.

Young, M. *et al.* (2014) 'Project, programme and portfolio maturity: a case study of Australian Federal Government', *International Journal of Managing Projects in Business*, 7(2), pp. 215–230. doi: 10.1108/IJMPB-08-2013-0034.

Zarrella, E. (KPMG) (2005) *Global IT project management survey: How committed are you*. Hong Kong, China. Available at: https://scholar.google.co.za/scholar?cluster=6428167755967665028&hl=en&as_sdt=2005&sciodt=0,5.

ZHANG, L. and LI, Y. (no date) 'CMMI/TSP/PSP', *pdfs.semanticscholar.org*. Available at: https://pdfs.semanticscholar.org/9b8c/0e70f1a5e26b7113ca933958b4b89157601c.pdf (Accessed: 31 May 2019).

Dear Participants,

My name is Ephraim Bogopa and I am a master's student at the Department of Applied Information System at University of Johannesburg. I am currently conducting research study entitle: The Influence Of Software Development Project Maturity Levels On Software Project Outcome under supervision of Prof. Carl Marnewick. The main objective of the study is to determine whether there is a correlation between the software development project success and software development project maturity level.

You have been selected to participate in the study because you have participated or worked within software development projects. Therefore, you are in a position to provide the information that will lead towards achieving the objectives of the study. The questionnaire consists five sections and it will take you minimum of 20 minutes and 30 minutes at maximum to complete it.

As the participant of the study, your participation is voluntary which means you can decline to participate in the study, you are free to withdraw in the middle of the study and there is no compensation that will be provided for participating in the study. All the responses will be kept confidential and synonyms and antonyms will be used when reporting the results of the study. This is done in order to ensure confidentiality of the participant of the study.

If you have any further information, you are free to contact the researcher on ephraimbogopa@icloud.com or you can call on: 0840219504 or the supervisor at cmarnewick@uj.ac.za. By completing the questionnaire, indicates that you gave consent to participate in the study. The project received ethical clearance from the CBE RESEARCH ETHICS committee at the University of Johannesburg, and the certificate is attached. You can access the questionnaire by clicking on this link: *http://take-survey.com/statkon/software_development.htm*

Thank you for participating in the study

Kind regards
Mr Ephraim Bogopa

Masters Student
Cell: +2784 021 9504
Email: Ephraim.bogopa@icloud.com

Appendix B: Questionnaire

Section A : PERSONAL INFORMATION

NO	Items	Which of the following best describes your current position within your organization?											
1	Position/ Job Title	Senior Manager	1	Software Developer/Programmer	2	Project Manager	3	Software Architect	4	Business analyst	5	Quality Assurer (Testers)	6
		Project Administrator	7	Data Scientist	8	Other (Specify):	9						
2	Length of Time/Work experiences	Less than 1 year	1	1 – 5 years	2	5 - 10 years	3	10 - 15 years	4	15 - 20 years	5	More than 20 years	6

Section B: Project Success and Failure

PROJECT OUTCOME OF IN THE LAST 6 MONTHS QUESTIONS:

Instruction

Please rate the following in terms of frequency (Rate from 1-5; 1 is Never, while 5 is Every time):

PROJECT OUTCOMES	QUESTIONS On average, please rate your projects in the last 6 months	Never	Almost never	Occasionally/ Sometimes3	Almost every time	Every time
Budget	1. The project was completed within or below budget	□	□	□	□	□
Quality	2. The product met the customer's specifications	□	□	□	□	□
	3. The customer is using the delivered product	□	□	□	□	□
	4. The customer was satisfied with the project	□	□	□	□	□
Time	5. The project was completed on time or earlier	□	□	□	□	□

6. On average, what was the size of your projects in the last 6 months? (Please select only one)

Very Small	□
Small	□
Medium	□
Large	□
Very Large	□

7. How many projects has your organization delivered over the last 6 months? (Please select only one)

0	☐
1-5	☐
6-15	☐
16-50	☐
50+	☐

8. To what extent do you agree that the following project success factors are critical for the success of your software development projects? (Rate from 1-5; 1 is Strongly Disagree, while 5 is Strongly Agree):

Category name	Software Development Factors	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
8.1. People related factors	Effective project management skills/methodologies (project manager)	☐	☐	☐	☐	☐
	User/client involvement	☐	☐	☐	☐	☐
	Support from top management	☐	☐	☐	☐	☐
	Good leadership	☐	☐	☐	☐	☐
	Committed and motivated team	☐	☐	☐	☐	☐
	Good performance by vendors/contractors/consultants	☐	☐	☐	☐	☐
	Skilled and sufficient staff	☐	☐	☐	☐	☐
8.2. Process related	Clear requirements and specifications	☐	☐	☐	☐	☐
	Clear objectives and goals	☐	☐	☐	☐	☐
	Realistic schedule	☐	☐	☐	☐	☐
	Realistic budget	☐	☐	☐	☐	☐
	Frozen requirement	☐	☐	☐	☐	☐
	Effective communication and feedback	☐	☐	☐	☐	☐
	Proper planning	☐	☐	☐	☐	☐
	Appropriate development processes/methodologies (process)	☐	☐	☐	☐	☐
	End-user training provision	☐	☐	☐	☐	☐
	Up-to-date progress reporting	☐	☐	☐	☐	☐
	Adequate resources	☐	☐	☐	☐	☐
	Effective monitoring and control	☐	☐	☐	☐	☐
	Risk management	☐	☐	☐	☐	☐
	Clear assignment of roles and responsibilities	☐	☐	☐	☐	☐
	Effective change and configuration management	☐	☐	☐	☐	☐
	Good quality management	☐	☐	☐	☐	☐
8.3. Technical related factors	Supporting tools and good infrastructure	☐	☐	☐	☐	☐
	Familiar with technology/development methodology	☐	☐	☐	☐	☐
	Complexity, project size, duration, and number of organisations involved	☐	☐	☐	☐	☐

Section C: Software Development Process Questions

Instruction:

Please read the following statements about software development processes and indicate to what degree do you agree that the process is implemented within your organisation (1= strongly disagree, 5 =strongly agree):

	SOFTWARE DEVELOPNET PROCESS AREAS	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
1	**REQUIREMENTS MANAGEMENT**	1	2	3	4	5
C1.1	The commitment to requirements is obtained from the project participants.	☐	☐	☐	☐	☐
C1.2	Project participants manages the changes imposed to existing requirements.	☐	☐	☐	☐	☐
2	**REQUIREMENTS DEVELOPMENT**	1	2	3	4	5
C2.1	During customer requirements development, the stakeholder needs, expectations and interfaces are collected by software development team.	☐	☐	☐	☐	☐
C2.2	The stakeholder needs, expectations and interfaces are translated into customer requirements exactly as required.	☐	☐	☐	☐	☐
C2.3	The customer requirements are refined and elaborated to develop software product and its component required.	☐	☐	☐	☐	☐
C2.4	The requirements are analysed and validated against risks in the early phases of the software projects.	☐	☐	☐	☐	☐
3	**TECHNICAL SOLUTION**	1	2	3	4	5
C3.1	Product or product component solutions are selected from alternative solutions.	☐	☐	☐	☐	☐
C3.2	Product or product component designs are developed.	☐	☐	☐	☐	☐
C3.3	Software product components designed are implemented.	☐	☐	☐	☐	☐
C3.4	Supporting documentation are implemented from their designs.	☐	☐	☐	☐	☐
4	**PRODUCT INTEGRATION**	1	2	3	4	5
C4.1	Preparation for product integration is conducted.	☐	☐	☐	☐	☐
C4.2	The product integration environment is prepared.	☐	☐	☐	☐	☐
C4.3	Product integration sequence is in place.	☐	☐	☐	☐	☐
C4.4	The product component interfaces are tested (both internal and external) for compatibility before starting with the integration activities.	☐	☐	☐	☐	☐
C4.5	The software development team make sure that the assembled product components are ready for integration.	☐	☐	☐	☐	☐
C4.6	The success of the integration is validated.	☐	☐	☐	☐	☐
C5.7	Packaging the assembled products components for the delivery to customer.	☐	☐	☐	☐	☐
5	**VERIFICATION**	1	2	3	4	5
C5.1	The work products that needs verification are identified.	☐	☐	☐	☐	☐
C5.2	Appropriate verification environment is prepared.	☐	☐	☐	☐	☐
C5.3	Establish verification procedures and criteria.	☐	☐	☐	☐	☐
C5.4	Peer reviews are performed on selected work products to ensure it meets specified requirements.	☐	☐	☐	☐	☐
C5.5	Selected work products are verified against their specified requirements.	☐	☐	☐	☐	☐

6	VALIDATION	1	2	3	4	5
C6.1	Preparation for validation is conducted.	☐	☐	☐	☐	☐
C6.2	The software team ensures the product meets its specified requirements.	☐	☐	☐	☐	☐
C6.3	A final test environment is close as possible to the environment in which the product or product components will be used/performing life.	☐	☐	☐	☐	☐

Section D: Project Management Maturity level Questions

Instruction:

Please read the following statements about project management processes and rate (indicate) to what extent do you agree that the process is implemented within your organisation (Rate from 1-5; 1 is Strongly Disagree, while 5 is Strongly Agree):

	PROJECT MANAGEMENT MATURITY PROCESSES	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
1	PROJECT PLANNING	1	2	3	4	5
D1.1	The estimates of project planning parameters (scope of the project, work, effort and cost required) are established.	☐	☐	☐	☐	☐
D1.2	A project plan is established as the basis for managing the project (e.g. plan for data management, needed knowledge and required skills).	☐	☐	☐	☐	☐
D1.3	Commitments by stakeholders to the project plan is obtained.	☐	☐	☐	☐	☐
2	PROJECT MONITORING AND CONTROL	1	2	3	4	5
D2.1	Actual project progress and performance against the project plan are monitored.	☐	☐	☐	☐	☐
D2.2	Corrective actions are managed to closure when the project's performance or results deviate significantly from the plan.	☐	☐	☐	☐	☐
3	SUPPLIER AGREEMENT MANAGEMENT (Only applicable to organisations which have external suppliers or contractors providing services)	1	2	3	4	5
D3.1	Agreements with the suppliers is established according to the types of acquisitions made.	☐	☐	☐	☐	☐
D3.2	The vendors qualified to supply the required types of products or product components are determined.	☐	☐	☐	☐	☐
D3.3	Both the project and suppliers are satisfied with the agreements.	☐	☐	☐	☐	☐
D3.4	The creation and delivery of the product is monitored.	☐	☐	☐	☐	☐
4	INTEGRATED PROJECT MANAGEMENT	1	2	3	4	5
D4.1	The project is conducted using a defined process tailored from the organisation's set of standard processes.	☐	☐	☐	☐	☐
D4.2	The collaboration between the project and relevant stakeholders is emphasized.	☐	☐	☐	☐	☐
D4.3	The coordination issues that might arise between relevant stakeholders and project teams are resolved.	☐	☐	☐	☐	☐
D4.4	A shared vision of the project is always ensured among individual teams.	☐	☐	☐	☐	☐

5	RISK MANAGEMENT	1	2	3	4	5
D5.1	A risk management strategy is in place to categorise typical and known risks.	☐	☐	☐	☐	☐
D5.2	The risks identified are analysed to determine their relative importance.	☐	☐	☐	☐	☐
D5.3	Risks are handled and mitigated as appropriate to reduce adverse impacts on achieving the objectives.	☐	☐	☐	☐	☐
6	QUANTITATIVE PROJECT MANAGEMENT	1	2	3	4	5
D6.1	Preparation for quantitative management is conducted by establishing performance objectives.	☐	☐	☐	☐	☐
D6.2	The process performance are managed in order to remain in line with the project objectives.	☐	☐	☐	☐	☐
D6.3	The selected subprocess performance of the project are measured and their results are analysed.	☐	☐	☐	☐	☐

Section E: Organisation level: Process Management Questions

Instruction:

Please read the following statements about Organisation processes and indicate to what extent do you agree that the process is implemented within your organisation (1= strongly disagree, 5 =strongly agree):

	ORGANIZATIONAL PROCESSES	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
1	ORGANIZATIONAL PROCESS FOCUS	1	2	3	4	5
E1.1	The organisation determines the process improvement opportunities that it will focus on (e.g. Improvements needs of software used).	☐	☐	☐	☐	☐
E1.2	The organisation assesses the process improvements needs.	☐	☐	☐	☐	☐
E1.3	The organisation identify process improvements targets.	☐	☐	☐	☐	☐
E1.4	The organisation has a plan to establish process action plans.	☐	☐	☐	☐	☐
E1.5	The organisation implement process improvements over time.	☐	☐	☐	☐	☐
E1.6	Organisational process assets are deployed across the organisation.	☐	☐	☐	☐	☐
E1.7	Process related experiences are incorporated into organizational process assets.	☐	☐	☐	☐	☐
2	ORGANIZATIONAL PROCESS DEFINATION	1	2	3	4	5
E2.1	The organisation establishes a standardized set of processes that teams within your organisation can access.	☐	☐	☐	☐	☐
E2.2	The organisation shapes standardised set of processes so that they can manage their work.	☐	☐	☐	☐	☐
3	ORGANIZATIONAL TRAINING	1	2	3	4	5
E3.1	A training capability is developed (resources and materials), which supports the organisation to deliver the courses.	☐	☐	☐	☐	☐
E3.2	The organisation develops skills and knowledge for its employees by offering the training.	☐	☐	☐	☐	☐
E3.3	The organisation collects feedback on training provided.	☐	☐	☐	☐	☐
4	ORGANIZATIONAL PROCESS PERFOMANCE	1	2	3	4	5
E4.1	The performance of the selected processes is recorded.	☐	☐	☐	☐	☐

E4.2	The performance measuring mechanics is established.	☐	☐	☐	☐	☐
5	**ORGANIZATIONAL PERFORMANCE MANAGEMENT**	1	2	3	4	5
E5.1	The organisation analyses the improvements opportunities and proposals.	☐	☐	☐	☐	☐
E5.2	The organisations is piloting the selected improvements.	☐	☐	☐	☐	☐
E5.3	The promising improvements are chosen for deployments.	☐	☐	☐	☐	☐
E5.4	Your organisation systematically deploys the improvements into the organisation.	☐	☐	☐	☐	☐
E5.5	The deployments are planned.	☐	☐	☐	☐	☐
E5.6	The improvements deployed are managed against the plan.	☐	☐	☐	☐	☐
E5.7	The organisation measures improvements results.	☐	☐	☐	☐	☐

OPTIONAL INFORMATION

If you would like to be informed of the final research findings, please contact Ephraim Bogopa on 0840219504 or ephraimbogopa@icloud.com.

APPENDIX C: ETHICAL CLEARANCE CERTIFICATE

CBEREC and SUBCOMMITTEES 2017

UNIVERSITY OF JOHANNESBURG

CBE RESEARCH ETHICS COMMITTEE

Dear M E Bogopa and Prof Marnewick

ETHICAL CLEARANCE GRANTED FOR RESEARCH PROJECT

This letter serves to confirm that the proposed research project has been granted ethical clearance by the School of Consumer Intelligence and Information Systems Ethics committee at the University of Johannesburg. Please refer to the report below for the ethical clearance number and specified conditions of approval.

ETHICAL CLEARANCE REPORT

Applicant	Moketo Ephraim Bogopa
Supervisor	Prof C Marnewick
Student/staff number	201477245
Title	THE INFLUENCE OF SOFTWARE DEVELOPMENT PROJECT MATURITY LEVELS ON SOFTWARE PROJECT OUTCOME
Decision date at meeting **Decision at Department / School** **Decision at College Meeting Decision** **at CBE REC**	16 July 2018 School of Consumer Intelligence and Information Systems Ethics Committee (Sub-committee of CBEREC)
Reviewers	SCiiS ethics committee members
Ethical clearance code	2018SCiiS 02
Rating of most recent application	CODE 01

CODE 01 -	Approved	CODE 02 -	Approved with suggestions without re-submission
CODE 03 -	Not approved, may re-submit	CODE 04 -	Not approved, no re-submission allowed

Page **1** of **2**

RESEARCH COMPLIES WITH	COMPLIANCE	NON-COMPLIANCE / DETAILS / RECOMMENDATIONS / CONDITIONS OF APPROVAL
The right to privacy, confidentiality and anonymity	Yes	The Researchers need to ensure participants' / research subjects anonymity is assured at all times
The right to equality, justice, human dignity/life and protection against harm	Yes	
The right to freedom of choice, expression and access to information	Yes	
Right of the community and science community	Yes	
The researcher will not experience any harm in conducting the research	Yes	
Informed consent/letters of request	Yes	

SCiiS

APPROVED

______16/7/2018__________

Research and Ethics Committee