The relationship between systems development methodologies and Information Technology project success

MS Nkone

12570850

Dissertation submitted in fulfilment of the requirements for the degree *Magister Scientiae* in Computer Science at the Potchefstroom Campus of the North-West University

Supervisor: Prof HM Huisman

May 2013

It all starts here ™

NORTH-WEST UNIVERSITY
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT

Abstract

The purpose of this study was to investigate the relationship between systems development methodologies (SDMs) and the success of Information Technology (IT) projects. The study also seeks to find other critical success factors (CSFs) that influence IT projects success.

What initiated this study, with reference to the literature review, is the apparent general view that IT project deliveries are still late, over budget, and unpredictable (Chow & Cao, 2008:961; The Standish Group, 2004). To some extent, the entire project fails before delivering an application. Hence this causes the need to investigate the employment of SDMs and their belief that SDMs improve quality in the development of IT projects and permit more flexible deployment to IT projects (Idea Group Publishing, 2006:13; Dyba *et al*., 2005:447; Mihailescu & Mihailescu, 2009:1). However, SDMs are still less popularly used (Siau & Tan, 2005:3132; Masrek I., 2008:137). The study provides insight into the relationship between SDMs and the IT projects.

A survey using a questionnaire was carried out to obtain the data. The study employed a positivist paradigm and used a quantitative approach. A total of 132 questionnaires were returned from systems developers, IT project managers and team leaders from system development companies.

It was found that there is a relationship between systems development methodologies (SDMs) and IT project success. Despite recognizing the benefits and advantages of using SDMs, some respondents disclosed that they were not inclined to employ SDMs and the popular reason was that their profile of development projects didn’t require the use of SDMs. Multiple regression analysis was used to determine the relationship. Consequently, the top ten CSFs were observed and it was found that “identifying potential risk” and “opportunity” were the most favourable factors.

The study could have obtained richer and more insightful information regarding SDMs and critical success factors if more data had been collected. Future work should seek to determine the risks, challenges and problems associated with the adoption of SDMs.

Keywords: Systems development methodologies, IT project success, critical success factors, systems development, Information systems.

Samevatting

Die doel van hierdie studie was om ondersoek in te stel na die verhouding tussen stelselontwikkeling metodologieë (SOM) en die sukses van Inligtingstegnologieprojekte. Die studie wou ook vasstel watter ander kritieke sukses-faktore IT-projekte se sukses kan beïnvloed.

Wat gelei het tot die studie is die algemene houding dat IT-projekaflewering steeds laat gebeur, begrotings oorskrei en onvoorspelbaar is (Chow & Cao, 2008:961; The Standish Group, 2004). Tot ‘n mate faal ‘n hele projek dus voordat ‘n toepassing eens gedoen is. Dit lei tot die behoefte om te kyk na die implementering van SOM's en die geloof dat SOM's kwaliteit in die ontwikkeling van IT-projekte bevorder en meer elastiese toepassing moontlik maak (Idea Group Publishing, 2006:13; Dyba *et al*., 2005:447; Mihailescu & Mihailescu, 2009:1). Dit is egter waar dat SOM's minder gewild is in die veld van toepassing (Siau & Tan, 2005:3132; Masrek I., 2008:137). Die studie verskaf insigte in die verhouding tussen SOM's en IT-projekte.

’n Oorsigvraelys is gebruik om die data in te samel. Die studie gaan uit van ’n positivistiese paradigma en maak gebruik van ’n kwantitatiewe benadering. ‘n Totaal van 132 vraelyste is voltooi en teruggestuur deur stelselontwerpers, IT-projekbestuurders en spanleiers uit stelselontwikkelingsmaatskappye.

Daar is bevind dat daar ’n verhouding is tussen stelselontwikkelingsmetodologieë en IT-projek-sukses. Ten spyte van ’n erkenning van die voordele van die gebruik van SOM's het sommige respondente aangedui dat hulle nie neig om SOM's te gebruik, en die algemeenste rede was dat hulle ontwikkelingsprojekprofiele nie die gebruik daarvan nodig het nie. Veelvuldige regressie-analise is gebruik om die verhouding te bepaal. Hieruit is die top-tien kritieke suksesfaktore uitgelig en daar is bevind dat die “identifisering van risiko” en “geleentheid” die sterkste faktore was.

Die studie kon moontlik ryker en meer insigvolle inligting oor SOM's en kritieke sukses-faktore opgelewer het as meer data gekollekteer kon word. Toekomstige werk in die veld moet kyk na ’n bepaling van die risiko's, uitdagings en probleme geassosieer met die aanvaarding van SOM's.

Sleutelwoorde: Stelselontwikkelingsmetodologieë, IT-projeksukses, kritieke sukses-faktore, stelselontwikkeling, inligtingstelsels

Acknowledgements

I would like to thank God for granting me His mercy and wisdom to accomplish this work successfully. The deeds of God's hands are faithful and just and His precepts are trustworthy (Psalm 111:7). Truly God is awesome and His promises are "Yes and Amen" (2 Corinthians 1:20). I will always love and serve Him with all my heart.

I also would like to extend my special thanks to my mother Mmalefu Alice Nkone, my sister Lerato Portia Nkone and my aunt Mmakgauta Maria Nkone, as the diligence and commitment I could apply throughout this pilgrimage were due to their prayers and support. They have always believed in me, even when my enthusiasm was at its lowest. During difficult times, they would be supportive to me and remind me of the fruits of perseverance in the face of hardship, and eventually rekindle my ambition. I am grateful to God for blessing me with such family whose hearts are deeply committed to Him.

Thanks are also due to my colleagues; Manini Kganakga for her encouragement and support, to Maria Mopelong who is also my spiritual mentor as her prayers and motivations have always made me remember that nothing is impossible with God and that for every pain there is a reward. My persistence and strength in this endeavour were as a result of their prayers and their availability in times of distress. I am grateful to God for them being part of my life

I would like to express my particular thanks to my supervisor Prof. Magda Huisman for her guidance, kindness and support. Her fascination with her profession is of a superior inspirational nature.

I also would like to thank Prof. Annette Combrink for the valuable language editing to an entire version of this dissertation, and express my gratitude to Dr Suria Elilis for the assessment and veracity of the statistical analysis and to Prof Casper Lessing for the verification of the reference list.

My great appreciation also goes to all the companies that were involved in this research project; my appreciation to their employees for the openness, time and effort they applied in expressing their views. In particular, I would like to mention

Martilene Orffer - CEO Psybergate in Cape Town, Cedric Vanderlinden – Synapsis Software in Port Elizabeth and Oriah Selolo - Central University of Technology in Bloemfontein, as their support and high level of commitment were exceptional.

As to myself, I am deeply grateful to God to for being His child whom He loves so much through Jesus Christ. I pray that all the fruits of the Spirit according to Galatians 5:22 (But the fruit of the Spirit is love, joy, peace, patience, kindness, goodness, faithfulness, gentleness, self-control) be apparent in my life, and that He may grant me the wisdom and power to pursue my career goals and His purpose for my life.

To all, I deeply and humbly express my thanks for the job well done. Keep up with the good work and May God Bless You.

Table of contents

Chapter 1: Introduction

Chapter 2: The impact of systems development methodologies and critical success factors to IT project success

Chapter 3: Research design

Chapter 4: Research results

Chapter 5: Discussion

List of Figures

List of Tables

ABBREVIATIONS

APM	Association for Project Management
AUP	Agile Unified Process
ASD	Analytical Software Design
CASE	Computer Aid Software Engineering
CATWOEA	Customers, actors, and transformation process, worldview, and owner, environmental
CCTA	Central Computer and telecommunications agency
CSFs	Critical success factors
DFD	Data flow diagram
DSDM	Dynamic systems development method
EEM	Enterprise Essential Model
ERD	Entity relationship diagram
ERP	Enterprise resource planning
ETHICS	Effective Technical and Human Implementation of Computer-based Systems Methodology
IEM	Information Engineering Methodology
IS	Information systems
ISD	Information systems development
IT	Information technology
JAD	Joint Application development
JSD	Jackson Structured Development
NLG	Natural language generation
OOD	Object Oriented design
OOHDM	Object-Oriented Hypermedia Design Method
OMT	Object modelling technique
PRINCE2	PRojects IN Controlled Environments
PS	Project success
RAD	Rapid application development
RMM	Remote management Module
RUP	Rational Unified Process
SADM	Several Software Architecture Design Methods

SDLC	System development life Cycle
SDMs	Systems development methodologies
SEM	System Essential Model
SQA	Software quality assurance
SSADM	Structured Systems Analysis and Design Method
SSM	Soft Systems Methodology
STRADIS	Structural Analysis, Design and Implementation of Information Systems
UML	Unified Model Language
W3DT	World Wide Web Design Technique
WISDM	Web Information Systems development Methodology
XP	Extreme Programming
YSM	Yourdon System Methodology

Chapter1: Introduction

The aim of this chapter is to provide an introduction about the particulars of the dissertation; the introduction is based on the role of systems development methodologies (SDMs) in the development of information technology (IT) projects. Furthermore, it presents the structure of the dissertation. The chapter is constituted as follows:

- Introduction
- Problem statement and substantiation
- Research aims and objectives
- The research layout
- Previous similar studies
- Conclusion

1.1. Introduction

The arena of information systems development (ISD) has gained popularity exponentially in the number of available information systems development methodologies (ISDMs) (Madsen *et al.*, 2006:225; Jain, 2010; Huisman & Iivari, 2006:29; Truex *et al*., 2000:54; Middleton & McCollum, 2001:9; Serra, 2002:2). Thus, the information systems development (ISD) literature often reports that information systems (IS) projects have experienced big failures (Madsen & Vidgen, 2009:3; Rodriguez-Repiso *et al.*, 2007:582; Chow & Cao, 2008: 961). Given this fact, the purpose of this study is to investigate the relationship between systems development methodologies (SDMs) and information technology (IT) projects' success. The study provides an overview of the impact of SDMs on the success of IT projects and furthermore determine the critical success factors (CSFs) related to the success of IT projects. CSFs are significant conditions which a project must achieve to be perceived as a success (Rodriguez-Repiso *et al.*, 2007:593)

It is believed that one of the best ways to produce effective and efficient information systems is through the application of systems development methodologies (Fowler, 2001:1). Hence, the use of an SDM is always regarded as the improvement of the quality and productivity in systems development. (Yahya *et al.,* 2002:15; Mihailescu & Mihailescu, 2009:1; Griffin & Brandyberry, 2008:2; Huisman & Iivari, 2006:33; Rowlands, 2004).

Thomas and Ferna´ndez (2008:734) declare that having a clear and well-defined insight into what the project has to achieve in reaching its goal would contribute towards attaining project success. Traditionally, all projects are goal-oriented and the achievement of a goal is determined by whether the project is a success or a failure. Thus, information systems are thought to be a success when they are finished on time, within budget and according to specifications (Karlsen *et al.,* 2005:526).

In view of that, developing an information system needs a sound systems development methodology in order to deliver proper and necessary functions on time, within budget, and at the desired performance level with acceptable quality to offer the minimum agreed functionality by using the assigned resources effectively and efficiently, and most importantly, be accepted by the intended end-users (Dalcher & Brodie, 2007:8).

With respect to the competence of actors, i.e. people using SDMs, it is assumed that highly experienced developers perceive the application of SDMs as less important; conversely, the developers assess it as of value to the profession (Omland & Nielsen, 2009:215). Consequently, developers possess the ability, they learn over time, they acquire the application domain knowledge, and they have some degree of autonomy and commitment, and they exercise personal motivation (Fitzgerald *et al.,* 2002:123-134). The study investigates and reports on these views.

Different types of systems development methodologies (SDMs) exist, but these SDMs do not adequately address the entire project as they vary from one project to another. Thus, more than one SDM may be used in different projects (Yahya *et al.,* 2002: 24).

As per the observation by Griffin and Brandyberry (2008:8) and Meso *et al.* (2006:18), the most widely used systems methodologies, according to their top ranking system, have been found to be SSADM, IEM, SSM, Yourdon, JSD, OOD; whereas RAD is considered to be less favourable, whereas according to the study by

Strode *et al.* (2009:6), among the useful ones XP is more popular while ASD is less used. These methodologies enable a greater flexibility in the development of information system processes. Different types of SMDs are also be examined and summarized in the study.

In addition, the study provides a framework to compare different SDMs. The comparison is based on the following elements and some of the elements are broken into sub-elements: philosophy (paradigm, objectives, domain and target) and model, tools, scope and practice (background and participants). The framework is constructed as per observations by Yaghini *et al.* (2009:9-10) and Avison and Fitzgerald (2008:604-613).

As per the study by Fitzgerald *et al. (*2002:xii*)*, the success in information systems development (ISD) is not based on any information systems method, nor on any implementation of the analysis and design techniques. Instead, there are a myriad of contextual factors attached to the enactment of the information system development process – this is the method-in-action which is assumed to determine the outcome of the development. This concept is further investigated in the study.

In spite of the aforementioned positive attributes about systems development methodologies, challenges exist (Rowlands, 2004). In this context, a recent study shows that systems development methodologies (SDMs) have not yet been popularly used in many organizations due to some complexities and ambiguities associated with its implementation (Huisman & Iivari, 2006:29; Kiely & Fitzgerald, 2005:2; Truex *et al.,* 2000:54),. Other challenges listed by Masrek *et al.* (2008: 144) include:

- Problems with SDMs
- Complexity
- Cost
- Difficulty in use
- Difficulty in understanding
- High level of detail
- Not widely known
- Too narrow/specific

- Obsolescence
- Broadness/inclusiveness

According to Yahya *et al.* (2002:27), the veracity of the following outlined problems that have always been discovered in the previous studies has been investigated, and it was found that they are still perceived as hiccups in the development of IS.

- Too much time is spent on understanding and learning the methodology (complicated methodology);
- The methodology does not cover the entire project development life cycle;
- The methodology is not applicable to the project development;
- There are limited numbers of skilled staff applying the methodology.

The study elaborates on these problems and challenges associated with the deployment of SDMs.

Furthermore, one among other critiques related to systems development methodologies is the fact that methods are seen as 'fitting' particular situations and the demand is for tailoring in order to accommodate the particular situation (Omland & Nielsen, 2009:215; Masrek *et al.*, 2008: 141). Similarly, Middleton and McCollum (2001:18) stipulate that the use of best method is misleading since projects and developers are perceived to be diverse. Thus, to a greater extent, formalized ISDMs are rarely applied in their entirety and exactly as originally conceptualized and as a consequence they are uniquely enacted by developers in work practice, i.e., they are adapted or tailored in a different way in any emergent development project (Fitzgerald *et al.,* 2002:13). More supplementary information about this matter is provided in the study.

1.2. Problem statement and substantiation

Recent studies show that many IT projects have failed due to budget and/or schedule overruns and/or because of not meeting users' requirements (Yeo, 2002:241). As such, the role of systems development methodologies to address these aspects has not received adequate attention (Yahya *et al.* 2002:17). Essentially, information technology professionals and stakeholders are interested in identifying the best critical success factors for development of information systems (Yeo, 2002:241; Rodriguez-Repiso *et al.*, 2007:593) in order to reduce these failures.

In particular, an early remedy for this dilemma was found within the academic areas of science and engineering, and therefore it was possible to propose the notion of SDMs which break up the development process into logical phases, i.e. analysis and design to advance the management of the development process of information systems (Kiely & Fitzgerald, 2005:2). The use and effectiveness of SDMs benefit the practices of the development of IT projects over a long period of time (Masrek *et al.*, 2008:139). Thus, SDMs consist of disciplined processes that make information systems more efficient and predictable upon development of IS projects (Fowler, 2001:1). Furthermore, Yardley (2002:83) also perceives information systems development methodology as an underlying factor in the IT industry; as it signifies many of the basic principles and procedures that organizations can implement in their choice of developing and deploying computer-based systems. Many systems are built with little thought or without the help of explicit systems development methodologies (SDMs) (Avison & Fitzgerald, 2008:23; Masrek *et al.*, 2008:137). The project team is adept at performing practical activities with little thought of processes (Fowler, 2001:1). As soon as enough information has been gathered, programming tasks commence. Henceforth, the emphasis is on programming aspects of development with little or no consideration of standard procedures (Kiely & Fitzgerald, 2005:2). Thus, practices of systems development methodology constitute the ambit in which to cultivate a more hands-on environment in collaborating with IT professionals and stakeholders, and consequently result in successful project delivery (Meto *et al.*, 2006:16). Nevertheless, software developers are reluctant for various reasons to use SDMs in the process of systems development (Toleman *et al.*, 2004:458).

The SDM environment has often changed in recent years; as such, a recent study reports that the use of SDMs is timely (Kiely & Fitzgerald, 2005:1). As a consequence, the CSF methods have been vigilantly observed in recent years (Chow & Cao, 2008: 962). It is therefore worthwhile to examine the relationship between systems development methodologies (SDMs) and the success of information technology (IT) projects. And further observe other CSFs which influence the success of the projects.

As we have learnt, the use of systems development methodologies (SDMs) has always been touted as the improvement of the quality and the productivity in the information systems development (Yahya *et al.,* 2002:15; Masrek *et al.,* 2008:143; Rowlands, 2004). They ensure that user requirements are met in order to produce high quality systems (McLeod, 2007:568). However, their evaluation is not keeping pace with the rapid growth of SDMs. As such, failure to evaluate the existing SDMs may lead to a lack of comprehension about the usefulness and effectiveness of the SDMs in organizations (Siau & Xin, 2005:3132).

It is worth noting that another aspect that is widely accepted as a challenge in the development of information systems is a failure to accommodate the human factor; however, systems development methodologies such as the Soft Systems Methodology approach recognize the importance of people in the development context and attempts to make sense of complex human activity systems which are typified by fussy or messy problem situations (Fitzgerald *et al., 2002:76)*. Yardley (2002:83) explains that methodologies such as SSADM, which are assumed to be the most popular all over the world, emphasize the formal and technical aspects of information systems development, overlooking the human, social, and organizational aspects, which are equally vital for achieving business success.

Despite the positive attributes the SDMs possess, many organisations spend millions of rands to advance information systems processes in order to overcome the pandemic information systems failures and challenges that are mostly experienced in organisations (Lu *et al.*, 2008:356; Akmanligil & Palvia 2004:45); Nonetheless, success of IT projects is still not being achieved due to delays and the higher than expected costs incurred during the information systems development (Akmanligil & Palvia, 2004:45). Another reason for IT projects delivery failure is the riddle of issue through the process of systems development (Jackson & Klobas, 2008:330).

In addition, Fitzgerald's (2008:569) contention is that the methodology includes the context of philosophy in which the user's opinions are taken into consideration. In particular, system development methodology is assumed to be an approach that acknowledges the following four components namely: philosophical approach, method steps/phases/levels, the process model, techniques and tools (Avison & Fitzgerald, 2008:34; Huisman & Iivari, 2006:32).

It is apparent that systems development methodologies (SDMs) have always been perceived to be beneficial and valuable in the arena of information systems (Middleton & McCollum, 2001:9). Empirical work on the use of SDMs is still limited (Yahya *et al.,* 2002:15) and their value has been questioned (Mihailescu & Mihailescu, 2009:1). Moreover, in general, the impact of various contextual factors on the development of software has not received much attention so far (Bern *et al.*, 2007:1). In view of that, there is a widespread conviction that not much has been done in advancing knowledge about the role of systems development methodologies (SDMs) in relation to IT Project success. As a result, the study will attend to the extent to which system development methodologies have an influence on Information Technology (IT) projects. Furthermore, the study will investigate the critical success factors that contribute to the success of the IT projects.

1.3. Research aims and objectives

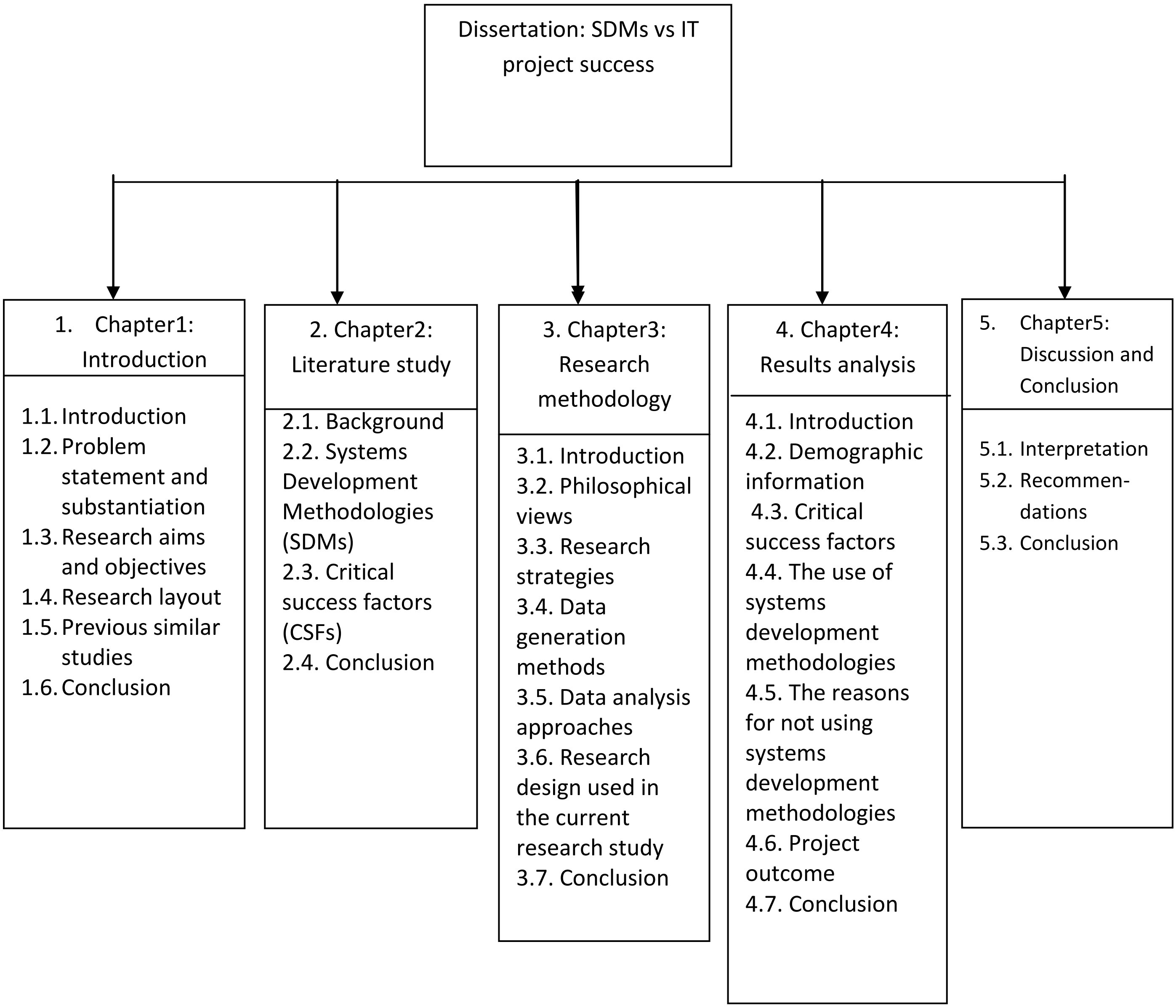
The purpose of this study is to examine the relationships between systems development methodologies (SDMs) and Information Technology (IT) project success. In order to achieve this, the following objectives were formulated:

- To investigate the use of systems development methodologies (SDMs) in information technology (IT) project success.
- To look at the extent to which systems development methodologies (SDMs) influence the success of Information technology (IT) project success.
- To determine other critical success factors that influence information technology projects success.
- To measure and compare the effect of systems development methodologies and other critical success factors on Information Technology (IT) project success.
- To evaluate the methodology that is mostly used in organizations.

1.4. The Research layout

Provisionally, the chapters of the dissertation are divided as shown in figure1.1.Consequently; to give an overall, the brief description of chapters is provided.

Figure 1.1: Research layout



1.4.1. Chapter1: Introduction

The purpose of this chapter is to provide an introduction to the role of SDMs in IT projects. To make sense of the research, the chapter presents background about the application of SDMs, the problem statement, research aims and objectives, previous similar study findings and the structure of the dissertation.

1.4.2. Chapter2: Literature study

The structure of this chapter has been arranged chronologically so that the reader can easily follow the role of CSFs and SDMs towards IT project success. Section one identifies and discusses the critical success factors (CSFs) that relate to the success of IT projects.

Different aspects linking SDMs and IT project success are discussed. Moreover, different types of SDMs are outlined and the way showed in which they are defined, from the aspects of the philosophical, method, process models and tools and techniques as being in harmony with the definition of systems development methodologies by Huisman and Iivari (2006:32) and Mihailescu and Mihailescu (2009:3). Hence, it compares them according to philosophy, model, tools, scope and practice.

1.4.3. Chapter3: Research methodology

The philosophical paradigm as well as the research approach used to carry out the study is a positivist approach achieved through a quantitative methodology, and a survey (using questionnaires distributed to public and private organisations) was carried out to obtain the findings. The results of this research have created knowledge regarding the relationship of SDMs and IT project success. Furthermore, the critical success factors relating to IT project success have also been measured. IT project managers or leaders, software developers, system analysts and information system developers were found to be appropriate candidates to participate in this study.

1.4.4. Chapter4: Results analysis

The results were obtained using statistical analysis as affected through the use of STATISTIKA version 10. The results of the survey were analysed using basic statistical measures such as mean comparison, frequency, factor analysis and reliability analyses.

1.4.5. Chapter5: Discussion and conclusion

The purpose of this chapter is to interpret and discuss the results of the analysis according to the research questions in order to fulfil the aims and objectives of this study. Eventually, the chapter presents the recommendations for advancement and future work pertaining to the study.

1.4.6. References

This section provides a list of previous authors used as references in this study. The reference style used is the abbreviated Harvard Style.

1.5. Previous similar studies

Table 1.5 presents the outcomes of previous studies regarding the usage and effectiveness of systems development methodologies (SDMs) in the project success of information technology (IT) projects. The investigations were based on studies from 2000 to 2009. The table displays the numbers, findings and the authors.

No	Findings	Author
1.5.1	The study shows that more than 50% of companies developing software employ some systems development methodologies	Griffin and Brandyberry (2008:8)
1.5.2	From the results obtained, 61.1% had agreed that the use of SDMs improved the information systems development; in that it could be completed on schedule or on time	Yahya *et al.* (2002:26)
1.5.3	The research suggests that competencies and SDMs deployed in a given development situation are intertwined in such a way that they cannot be separated in practice	Omland and Nielsen (2009:222)
1.5.4	It was recognized that RUP methodology together with PRINCE 2 methodology complemented each other and allowed any project management team to improve the control of the entire process and allow for more efficient change	Lancaster University (2005:25)

	management.	
1.5.5	It is suggested that the main purpose of SDMs is to advance human capabilities which would allow human agents to perform valuable functions. Thus, lack of attention to all these three components, i.e. structure, agency, and cultural system, will reduce the success of SDMs intervention.	Mihailescu and Mihailescu (2009:6)
1.5.6	SDMs are thought of as a guide to organizations for purposes of the achievement of the vision rather than a prescriptive basis for project planning and action.	Madsen *et al.* (2006:237)
1.5.7	It was found that the perceptions of IS managers of systems development methodologies were more positive than those of developers.	Huisman and Iivari (2006:40)
1.5.8	Methods seem more like idealizations than prescriptions, and might better be presented as “cases” or “exemplars” instead of practical frameworks.	Truex *et al.* (2000: 74)
1.5.9	The study suggests that effectively matching SDMs to an application domain does leverage knowledge management outcomes by influencing the effectiveness of knowledge-work processes in a systems development project.	Meso *et al.* (2006: 26)
1.5.10	Methods are no more applied blindly to the development process to ensure the success of project development process; instead they are tailored to fit the particular development project.	Kiely and Fitzgerald (2005:12)
1.5.11	The study found that an organization is probably unwise to use heavily prescriptive SDMs to improve its software development performance.	Middleton and McCollum (2001:18)
1.5.12	It is determined that organizational culture is a factor affecting successful adoption of an agile method	Strode *et al.* (2009: 7)
1.5.13	It was observed that levels of methods and process formalization constitute one of the contextual factors within company infrastructure categories that influenced software development practices.	Bern *et al.* (2007:8)
1.5.14	The findings have indicated that the organization use most of an in-house development SDMs in their systems development projects.	Masrek *et al.* (2008:144)
1.5.15	Practitioner experience suggests that agile methods are particularly suitable for projects where requirements are more abstract and difficult to define; thus, such organisations have either not adopted or moved away from traditional approaches	Toleman *et al.* (2004:469)

Table 1.5: Previous studies about SDMs

1.6. Conclusion

The objective of this chapter has been to present an introduction to the dissertation. The chapter is first characterized by providing an introduction which reveals the contextual correlation between the systems development methodologies (SDMs) and Information technology (IT) project success as per previous studies.

Furthermore, the chapter provides the problem statement in which it is believed that organizations are faced with challenges in information systems development (Chow & Cao, 2008:961). According to the survey, IT projects experience 40% of failures, and the average cost of these projects each year is believed to run to 1 million U.S. dollars (Lu *et al.*, 2008:356); hence, it is vital to identify and classify critical success factors (CSFs) impacting on ISD (Yeo, 2002:241). To some extent, SDMs are assumed to be one the CSFs (Yahya *et al.,* 2002:15) though they are not popularly used due to their complexities (Masrek *et al.*, 2008:137). As a result, the study examines its usage and effectiveness in the arena of ISD.

The research aims and objectives have been outlined. They are formulated on the basis of the research questions identified during the initial phase of the study. It is expected that the study will address these questions appropriately in order to achieve the purpose it was intended for.

Subsequently, the structure of the dissertation or the layer of the research is displayed in Figure 1; it comprises the following chapters and sections where the brief description has been provided such as the introduction, literature study, research methodology, results analysis, discussion, interpretation and references.

Lastly, Table 1.5 represents the outcomes of the previous studies about the effect and usage of SDMs to the success of IT projects. Fifteen studies have been quoted of which the author and the findings of the research are provided.

Chapter 2: Literature study:

2.1. Introduction

The chapter is divided into two sections. Section one is aimed at providing a discussion of the relationship between systems development methodologies and information technology (IT) project success. To reflect the logical cycle about the relationship between systems development methodologies and the success of IT project, the following aspects will be discussed:

- Background;
- The definitions of systems development methodologies;
- The usage and effectiveness of systems development methodologies;
- The advantages and benefits of systems development methodologies;
- The disadvantages and criticisms of systems development methodologies;
- Types of systems development methodologies such as JSD, SSADM, IEM, SSM, RUP, YDM, ETHICS, XP, STRADIS and WISDM; and
- A comparison framework of different types of systems development methodologies.

The purpose of section two is to investigate the critical success factors (CSFs) which are assumed to have a positive effect on the development of information technology (IT) project success. The section will present an overview discussion of the following aspects:

- Background;
- The list of critical success factors (CSF) per author's opinion;
- The ranking of success factors per frequency of citations from the literature;
- A review of subsequent years of The Standish Group's reports; and
- A discussion of critical success factors gleaned from the literature.

Section One

2.2 A discussion of relationship between systems development methodologies (SDMs) and IT project success

2.2.1 Background

The aim of this section is to present knowledge about the relationship between systems development methodologies (SDMs) and the success of information technology (IT) projects in organizations. According to Avison and Fitzgerald (2008:34) systems development methodology originated in 1970 from the earliest variant of the systems development life-cycle (SDLC) proposed in the UK by the National Computing Centre in the late 1960s. The purpose was to suggest good standards to build successful computer applications. In addition, the systems development methodology concept was provoked during the 'software crisis' discussion at the landmark conference which was held in Garmisch in Germany in 1968. From this conference there was a widespread decision to implement more systematic approaches to systems development (Fitzgerald *et al.,* 2002:22). In view of that, information systems methodologies have a vital role to play in order to deliver successful IT projects and it is within this scope that many of the elementary issues affecting IS project failure lie (Yardley, 2002:103).

2.2.2. Definitions of systems development methodologies

The purpose here is to provide some perspectives on systems development methodologies (SDMs). To aid in the understanding of various views of SDMs, a definition of terms from different researches will be presented.

According to Fitzgerald *et al. (*2002:5*)* from time to time methodology is used interchangeably with method, however, Avison and Fitzgerald (2008:569) have argued that the term 'methodology' is not similar to method as it has certain characteristics that are not implied by 'method', for example, the methodology includes the context of philosophy in which the user's opinions are taken into consideration. Eventually, it was concluded that *methodology* is a broader concept than *method*. Hence, Fitzgerald *et al. (*2002:5*)* use the term 'formalized method' to refer to those commercial, brand-name and the in-house methods which are formally documented.

Table 2.2 presents the definition of systems development methodologies as per pervious relevant studies. It provides the name of the author, a definition of SDMs and the key terms from the definition.

Author	Definition of SDM	Key term
Fitzgerald (2008:568)	"A systems development methodology is a recommended means to achieve the development, or part of the development, of information systems based on a set of rationales and an underlying philosophy that supports, justifies and makes coherent such a recommendation for a particular context. The recommended means usually include the identification of phases, procedures, tasks, rules, techniques, guidelines, documentation and tools. They might also include recommendations concerning the management and organization of the approach and identification and training of the participants. **"**	Recognizes phases, procedures, tasks, rules, techniques, guidelines, documentation and tools
Laudon (2000:383)	Collection of methods, and they are achieved within the different stages of systems development project for every activity performed. Systems development methodologies are mainly used to help to analyse, design, document and implement during the development of Information systems.	Analyse design, document and implement during the development of Information systems.
Vidgen *et al.* (2002:30)	Useful epistemological frameworks that can be drawn on during the process of systems development.	Epistemological frameworks
Meso *et al.* (2006:15)	The vehicle in which the systems development team finds a way to close the gap to the problem-solution in order to produce an effective business IS application for the problem that initiated the development effort.	Provides effective business to the IS applications
(Futrel *et al.* 2002:107; Meso *et al.* 2006:15)	Systems development methodologies serve as map that guides all project stakeholders to move forward and help them to understand whether they are making progress in a project.	Serves as guide to projects stakeholders
Yardley (2002:102)	Systems development methodology is an approach that offers the process to support the business systems development.	Supports the business systems development.
Truex *et al.* (2000:54)	Systems development methodologies are the mainstream of information system discourse and as the endeavour to pursue order, and of course, the predictable and universal approach to which information systems are developed. Method is	Predictable and universal approach to develop information

	conceptually inferred to be a process based rather than representation-based.	systems
Huisman and Iivari (2006*:*32)	Systems development methodologies may be described as the approach that constitute the four fundamental components; philosophical approach, phases, process model, and tools and techniques	Embraces philosophical approach, phases, process model, and tools and techniques

Table 2.2: Definitions of Systems Development Methodologies (SDMs) from different authors

In particular, to make clear the perception of SDMs, the following description will be the focal point of this study.

The systems development method may be described as the approach that embraces four fundamental components namely*:* Philosophical approach, Phases, Process Model, and Tools and techniques.(Huisman & Iivari, 2006*:*32) (see Figure 2.2).

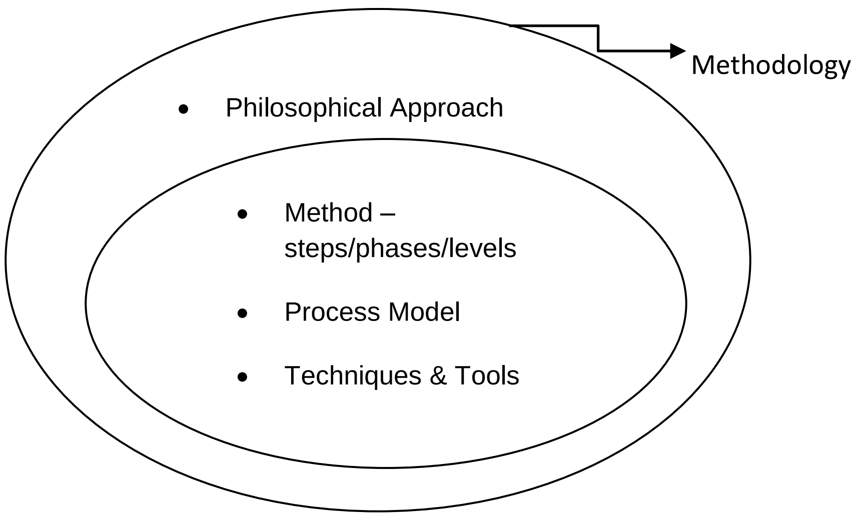


Figure 2.2: Components of systems development methodology

- Philosophical approach – It involves people's assumptions upon which the methodology is built, and characterizes development approaches such as structured, object-oriented and information modelling (Huisman & Iivari, 2006*:*32). Similarly, it is a set of opinions or way of thinking about target information systems (Avison & Fitzgerald, 2008*:*34).

- Method – this includes the levels or steps needing to be carried out as a sequential process. It is a systematic way that is established on the basis of a particular philosophy guided by the emergent system (Avison & Fitzgerald, 2008*:*34). Method constitutes a set of guiding principles, beliefs and values, set of goals and system processes that force interpretations and actions, and examples include*:* OMT, IE, etc. (Huisman & Iivari, 2006*:*32).
- Process Model – this is the logical analysis that describes what an emerged system wants to achieve and how it is going to achieve it (Avison & Fitzgerald, 2008*:*34). As a consequence, it reflects the sequence of steps in the course of systems development. Some of the examples are linear life-cycle and spiral models (Huisman & Iivari, 2006*:*32).
- Tools and Techniques – techniques help to ensure thorough decision-making in finding the necessary design in the development of information systems. Examples include DFD, ERD, etc. Therefore, tools are perceived to be mechanisms aimed at helping to develop information systems e.g. software packages (Avison & Fitzgerald, 2008*:*34).

In conclusion, it is evident from the aforementioned definitions that SDM is one of the fundamental issues of information systems to be applied during the development. It is commonly defined as an attempt to attend to complex and difficult activities in trying to compose the emerged information system. Furthermore, SDMs are defined as the components aimed at rationalising and effecting better control in the process of the development of information systems.

2.2.3. The use and effectiveness of SDMs in organizations

The use and effectiveness of systems development methodologies will be discussed through the following concepts namely; usage, adoption, in-house versus commercial methods, managers versus developers and non-use of SDMs.

2.2.3.1. The use of systems development methodologies

Although the methodological practices have been popularized in the academic research domain, businesses have not been able to universally embrace them (Griffin & Brandyberry, 2008:2). It is worth pointing out that the use of systems development methodology is not compulsory in organizations, hence practitioners are not persuasive in employing them during the development of IT projects (Pieterse, 2006*:*IV). As a result, the adoption of systems development methodologies by practitioners has been somewhat slow (Laudon, 2000*:*388).

In practice, not many software development organisations use SDMs, the common perception from previous studies being that many software developers are reluctant, for some reason, to use or adopt systems development methodologies (SDMs) in the course of developing IT projects (Laudon, 2000*:*388). The opposite is true with the study by McLeod (2007*:*566), where it shows that the measure of systems development methodology usage is more advanced than previous empirical studies might suggest. The study further discloses that, in the projects where SDMs were not used, it was as a result of the nature of projects (e.g. the project was small or non-critical, was a packaged solution involving little or no customization, or control of the project was outside the IS function).

In most cases, choices about the adoption of the standard methodology must be made by the IS department. However, the business managers should be the ones to make such decisions, as they are the ones to make investigations in terms of money, effort, time and ultimately business decisions (Avison & Fitzgerald, 2008:574). In view of that, organizations adopting the methodology are spending enormously as the cost includes training of staff and users, additional hardware and software, ongoing consultancy cost that might rise up to the level of the initial cost of the actual methodology (Avison & Fitzgerald, 2008:574).

In similar vein, Fitzgerald *et al.* (2002:97) are of the opinion that the choice that organizations are making with regard to the usage of the methodology in a given project depends on pragmatic reasons rather than the fundamental features of the method itself. This leads to the widespread observation that methodology users to a greater extent have different perspectives in the usage or adoption of systems development methodologies (SDMs) (Huisman & Iivari, 2006:29).

The research conducted by Pieterse (2006*:*IV) indicates that organizations adopt the systems development methodology due to the following reasons:

- A financial increase is acquired;
- there is a deficiency in knowledge;
- there are time limitations; and
- methodologies are not popularly applicable as yet.

In addition, Saini *et al.* (2009:88) outlined the following reasons advanced to justify the use of systematic development methodologies*:*

- They make it easier to understand systems by making a separation between conceptual design, logical design, and physical design;
- They design systems so that they are extendable and easily maintained;
- They move toward automated design tools and dynamic page generation;
- They are easier to use in order to manage development in the form of costs, time, task allocation, deliverables, etc.;
- They overcome reliance on designers who constructed the system;
- They are used to reduce risks associated with shortcuts and mistakes;
- They are used to produce documentation that is consistent from one project to the next; and
- Finally, a word of caution should be expressed by considering that:
    - systems development projects are not homogenous by nature;
    - there is not a single-best one-size-fits-all methodology; and
    - in practice, most organizations are not committed to any single methodology, and mix and match as appropriate.

2.2.3.2. The adoption of systems development methodologies

Though the choice of methodologies made by organizations depends on the particular needs they experience, yet to some organizations the methodology adopted still needs to be amplified by writing the detailed manual according to the requirements of their development staff (Avison & Fitzgerald, 2008:574).

In real development practice, formalized ISD methods are mostly not used in their entirety, nor as they were initially aimed for by their creators, even though they offer the template to guide the development practices. This brings into play the notion of method-in-action which is inferred to be uniquely enacted by the developer to meet any development project goal (Fitzgerald *et al.,* 2002:13).

In contrast, it is tempting to assume that one methodology can be universal, in that a single methodology can be applied to all kinds of projects throughout the organisation. Linked with an inappropriate recognition of developer-specific factors, IS development methodologies have always been deemed to become the 'one size fits all' solution to the IT project in the organisations (Vidgen *et al.*, 2002:4).

To rectify this misconception, more than one systems development methodology may probably be used to accomplish a project goal in the organization; in which, practically speaking, a methodology could be adopted in a particular phase of a project (Yahya *et al.*, 2002:25). However, the same methodology will not be interpreted and applied in the same way by different developers; neither will the same developer apply the same systems development methodology in the same way in more than one emergent project or in different development situations (Fitzgerald *et al., 2002:13*). Generally, systems development methodologies have different levels of complexity and rigour (Griffin & Brandyberry, 2008:2).

Consequently, this triggers the decision as to whether to use in-house or commercial methods which are discussed in the subsequent section.

2.2.3.3. In-house versus commercial methods

With regard to commercial SDMs and in-house methodology usage, it has been determined that most often organizations use in-house methods as an alternative, and these methods may be adapted from some form of commercial method (Akmanligil & Palvia, 2004:45**;** Masrek *et al.*, 2008:144; McLeod, 2007:567). The reason for this is that the in-house methodologies are perceived to be a better fit for the developmental context requirement's need and are more flexible than commercial methodologies (Kiely & Fitzgerald, 2003:10; Fitzgerald *et al.*, 2002:166**).**

Of those organizations using a formal methodology, many use methodologies by engaging in contracts with consulting companies; whereas others develop their own in-house methodologies to suit their needs, sometimes using other methodologies as a template or guide. Certainly, the choice as to whether any one or a specific methodology would be used will be determined by the condition of buyouts or mergers (Griffin & Brandyberry, 2008:2).

2.2.3.4. Managers versus developers

Systems development methodologies reflect management's agenda in the sense that IS managers are more pessimistic regarding the support and positive effect that systems development methodologies provide than it seem with systems developers (Huisman & Iivari, 2006:41). Seemingly, the perceptions of IS managers of systems development methodologies are more positive than those of the developers (Huisman & Iivari, 2006:33). Another point to consider is that with high-power distance, top executives or senior partners may exercise power to determine the way tasks should be carried out (e.g., possibly contrary to any IS development methods, or adjust them to suit any given situations). Conversely, with low-power distance, less authorized employees may feel empowered by the use of systems development methods, because they are less likely to be overruled (Fitzgerald *et al.,* 2002*:*128). Typically, Fitzgerald *et al. (*2002*:*128) hold the contentious view that commercial methods are more likely to be useful to inexperienced developers as the standard methods in-house are often favoured by experienced developers.

2.2.3.5. Reasons for not using systems development methodologies

While there are a number of significant arguments in the favour of systems development methodologies, there are also a number of arguments and pressures that question the use of such methodologies. Fowler's (2001:1) contentious issue is that, although the methodologies have existed for a long time, they have not been notable for being terribly successful; and little has been known about them. Moreover, methodologies are often heavy and finely detailed and this can appear to obstruct the productivity (Griffin & Brandyberry, 2008:2). As a result the nature of ISDMs (i.e. difficult, too abstract, too many deliverables to be generated) prevents them from been widely used in the organizations, hence, it is vital to look at criteria of the methodology prior to attempting to use it. Clearly, for a methodology to be feasibly applied, it should be easy to understand, and also be appropriate to be used in any types of cases (complex, big or small systems) and also be relevant across the development life cycle (Yahya *et al.,* 2002:31).

In summary, it is apparent from the aforesaid realities that the use of SDMs has always been considered as capable of improving the quality and productivity of systems development (Yahya *et al.,* 2002:15; Masrek *et al.,* 2008:143). Thus the efficiency of the development team and the quality of the developed product are increased. Nevertheless, there is little empirical evidence that can prove these assumptions as the systems are still overwhelmed by challenges of cost-overrun, being time-consuming, and failure to meet the users' requirements (Fitzgerald *et al.,* 2002:1).

To further untie the different perceptions of different researchers about the existence of systems development methodologies, the next sections seek to find the advantages of systems development methodologies that have been derived from organizations.

2.2.4. The advantages and benefits of systems development methodologies

In this sub-section, the advantages and benefits of systems development methodologies will be discussed through the following aspects: better end-product, better development, standardized process and procedures, supporting tool, knowledge and education, communication, systematic guide and benefit.

2.2.4.1. Better end-product

Systems development methodologies (SDMs) are perceived to be an attempt to improve the quality of the end-product of the information systems development process (Mihailescu & Mihailescu, 2009:1; Griffin & Brandyberry, 2008:2; Huisman & Iivari, 2006:33; Yahya *et al.,* 2002:15; Masrek *et al.,* 2008:143). As a result, systems development methodologies are aimed at meeting the system's requirements, completely within the budget and the schedule (Avison & Fitzgerald, 2008:570; Yahya *et al.,* 2002:15; Saini *et al.,* 2009:89; McLeod, 2007:563).

According to Avison and Fitzgerald (2008:570) people have the need to have better information systems; therefore, they use systems development methodology in an attempt to improve the end product of the development process. The components of the quality of an information system include acceptability, availability, cohesiveness, compatibility, documentation, ease of learning, effectiveness, efficiency, fast development rate, flexibility, functionality, implement ability, low coupling, amenability, portability, reliability, robustness, security, simplicity, testability, timeliness and visibility. The maximization of these criteria could potentially lead to a better product during the development of Information systems (Avison & Fitzgerald, 2008*:*570).

2.2.4.2. Better development

According to Yahya *et al.* (2002*:*26), systems development methodologies are also helpful in the development of an information system in that they involve certain types of tools such as CASE tools, word-processing, spreadsheets, graphic tools, presentation software and charting tools. Thus, tools tend to ease the project development. Similarly, information systems development methodologies bring together various, often vendor-specific procedures, techniques, tools and documentation aids relevant to different sections of the information systems development life-cycle (Nandhakumar & Avison, 1999*:*176; Mihailescu & Mihailescu, 2009:2).

Avison and Fitzgerald (2008*:*570) state that there are outputs (deliverables) likely to accumulate in every phase of a systems development methodology during the information systems development. It is assumed that since the emergence of systems development methodologies has become apparent, the levels of skills required of professionals have been reduced, and observably, the costs are expected to be reduced while the development of information systems through the implementation of systems development methodologies would be improved.

2.2.4.3. Standardizing processes and procedures

Typically, methodologies entail the standard processes that aid project managers in achieving the goals of the project. Nonetheless, the use of systems development methodologies differs from one organization to another; hence, it is vital to discern how organizations feel about the adoption of a systems development methodology (Yahya *et al.,* 2002:15).

During the development system, a standard and common process is followed, and this makes it easy to integrate systems. Thus staff members become familiar with the process and this accrues to the benefit of staff members in gaining skills and knowledge and consequently makes it easy for them to change from project to project without being retrained (Avison & Fitzgerald, 2008:570).

According to Futrel *et al.* (2002:107) a standard for IT systems of the Federal Republic of Germany comprises reasons for the importance of a standardized process. The purpose of these processes is to achieve the following objectives:

- The advancement and the assurance of the quality;
- The successful results to be delivered might be guaranteed by the standardized procedures; and
- The cost for the implementation of methodology can be easily checked.

It is these standard procedures that enable the costs calculation to be viable and visible and aid to recognize any risks that could emerge, in that they minimize the use of resources and reduce friction losses among stakeholders; as a result, progress of the project is liable to be effectively monitored (Futrel *et al.,* 2002:107).

The other point to make is that standard procedures reduce misunderstandings among all stakeholders, as the requirements and the needs are effectively communicated and agreed-upon. As a result, all parties involved will be efficiently supported through their underlying needs (Futrel *et al.,* 2002:107).

Pieterse (2006:IV) came up with the idea that the developers are of the opinion that a systems development methodology produces a system of high quality. Therefore, it proves that the systems development methodology has major strengths such as consistency and well-being structuring. Hence it relies on certain sets of procedures and steps to be followed when implemented.

2.2.4.4. Supporting tool

A systems development methodology is deemed to be one of the most essential supporting tools to achieve complicated practices that comply with the preface of information systems development (Yahya *et al.,* 2002:26). Hence, the role of the SDM is assumed to be supporting the potential stakeholder in meeting their requirements and helps managing or solving the problem during the systems development (Mihailescu & Mihailescu, 2009:3).

2.2.4.5. Knowledge and education

Certainly, SDMs serve as an instrument to transfer knowledge between experienced and novice developers and templates to guide the development practice of newly-emerged systems (Fitzgerald, 2002:120). Thus, methodology in use also serves as the framework for the knowledge-work processes that underpin the business IS application with which the team is involved (Meso *et al.,* 2006:15), hence education and training are adhered to as the benefit in systems development methodologies (Huisman & Iivari, 2006:41). As a consequence, as an individual unit of adoption, ISDM is perceived to be a potential means which, if used, improves job performance within the business. The deployment of ISDM is perceived to improve the career of individual developers, and cater for support or transfer of knowledge base within the organization or community (Mihailescu & Mihailescu, 2009:4).

2.2.4.6. Communication

Furthermore, systems development methodologies allow feasible communication among all stakeholders (Futrel *et al.,* 2002:107; Mihailescu & Mihailescu, 2009:5). Effectively, the use of systems development methodologies improves the communication between developers and users. (McLeod, 2007:568**).** In particular, XP is another type of methodology which underpins the communication and coordination among the team members to a greater extent; hence it promotes the teamwork spirit among the stakeholders (Qumer & Henderson-Sellers, 2008:283).

2.2.4.7. Systematic guide

In practice, many consulting companies have developed and implemented various commercial systems development methodologies that offer systematic guidance during the development of information systems (Leem & Kim, 2002:249; Meso *et al.,* 2006:17). Notably, standard methods offer detailed guidelines that inform the action of developers and the specifications for the tasks required to be carried out during the systems development (McLeod, 2007:563).

Certainly, one of the major reasons for IT projects to fail to meet the requirements is because project managers operate reactively rather than proactively. When a problem appears, they are hesitant to make effective and viable decisions, which often results in a situation that could destroy a project. It is assumed that this behaviour occurs due to a lack of guidelines to aid in the development process. In view of that, practices of systems development methodology are believed to serve as a guide necessary for the development of the project (Avison & Fitzgerald, 2008:574), and as a result methodology is recommended to deliver proper and required functions, on time, within budget, and that meet user expectations (Wetherbe, 2001:604).

2.2.4.8. Benefit

According to Saini *et al.* (2009:89), the use of systematic development methodologies can be of great value to companies in that:

- It reduces cycle time and / or reduced maintenance;
- It increases productivity;
- There are higher quality systems; and
- It improves project management; time and cost estimates, tracking and resource utilization.

In the study by Masrek *et al.* (2008:143) it was found through empirical evidence that the use of systems development methodologies provides benefits in the development of information systems. The following benefits are arranged from the highest ranking to the lowest ranking*:* The systems development methodology was perceived to:

- Allow better project control;
- Facilitate communication among developers;
- Ensure that systems meet user needs better;
- Increase visibility of systems development process;
- Improve the quality of the systems developed;
- Help ensure that documentation is produced;
- Facilitate interchangeability of developers among projects;
- Increase the likelihood that systems will be delivered on time;
- Reduce maintenance of systems; and
- Increase the likelihood that systems will be delivered within budget.

Moreover, the survey by Yahya *et al.* (2002:15) discovered the following benefits when carrying out information systems development projects in the adoption of methodologies.

- Higher quality of produced documents;
- Increases in the involvement of the computer user;
- Increases in the number of information systems products;
- A reduction of the maintenance cost;
- A reduction of the design error;
- Fulfilment of the user's requirements; and
- An improvement of communication between the user and the systems developer.

The significant feature of systems development methodologies is that they provide a reminder that triggers knowledge and act as guides to interpretation and action in IS development projects, thus promoting communication, coordination, control, and production (Mihailescu & Mihailescu, 2009:5). With such conceptualization, the claim that SDMs are aimed at improving the development of information systems in terms of productivity and quality becomes apparent, nevertheless, there is little evidence to prove the veracity of such a remark (Yahya *et al.,* 2002:15). As a consequence, this controversy invokes the proposal of the next section in which the disadvantages and critics of systems development methodologies are discussed.

2.2.5. Disadvantages and criticisms of systems development methodologies

In this sub-section, disadvantages and criticism of systems development methodologies will be discussed with reference to their being time-consuming, with complex development, a need for method tailoring and not being widely practised as well as lacking full control.

2.2.5.1. Time-consuming during development

According to Truex *et al*. (2000:54), though systems development methodologies for developing information systems have always been reported as being significant in the information systems discipline, there are still problems associated with their practicality. The main problems are that systems take time to develop, there is an overrun of cost, and the initial requirements for the system are not met.

In the early days, as envisaged, SDMs were designed to address the complexities and difficulties during the process of developing information systems. Nevertheless, the inherent problem was that the system still failed to be developed on time, within the budget, or be of good quality (Griffin & Brandyberry, 2008:2; Fitzgerald *et al.,* 2002:3).

In view of that, Wetherbe *et al.* (2001:629) state that methodology application often takes time to complete, hence they operate well in large projects with clearly-defined requirements where there is no pressure to complete the project quickly. According to the study conducted by Pieterse (2006: IV), the results indicated that the use of systems development methodology in organizations is time-consuming, and as a result, low-quality systems are taken out in order to gain a faster delivery time.

2.2.5.2. Complex development

Methodologies are often considered to be process-heavy, and are likely to obstruct productivity due to their finely detailed requirements (Griffin & Brandyberry, 2002). Due to such step-by-step techniques and the fact that every step must be completed before the next one can commerce, the development may be delayed. Hence, organizations are still hesitant to apply systems development methodologies during information systems development (Laudon, 2000:388). This complexity results in the entire development process not being fully successfully accomplished as per purported objectives (McLeod, 2007:569).

In effect, the study by Yahya *et al.* (2002:31) ascertains that, even though the methodology is used in organizations, because of the perception attached that they improve productivity, quality and communication; yet they are not uniformly accepted due to their complexity (i.e. difficult, too abstract, too many deliverables to be generated). Practically, inconsistency makes the process analysis of methodologies unreliable, thus making it difficult to reach the process improvements (Dahiya & Jain, 2010).

Furthermore, Dahiya and Jain (2010), assume that it is of the utmost importance to follow the processes defined in the methodologies in a consistent and accurate manner across all projects, thus ensuring the effective use of methodologies. However, these become challenging tasks due to:

- Software-development processes being complex, with their step-by-step techniques of interdependent activities (Griffin & Brandyberry, 2002; Dahiya & Jain, 2010). Most of the processes and methodologies bring about reference documents only. As a result, it makes it difficult for practitioners to gain knowledge about the processes from the document references and follow them in their projects (Dahiya & Jain, 2010).
- Practitioners follow the processes manually and deliver the necessary data into various life-cycle tools that do not have much integration with the engineering tools. These different sets of tools hardly allow an accurate methodology implementation across multiple projects, and this generates data and reports inconsistent with the organization (Dahiya & Jain, 2010).

2.2.5.3. Need for method tailoring

It is apparent from the aforementioned statements that systems development methodologies (SDMs) do not fully satisfy organizational requirements and they are not suitable to every development project even if they are justified particularly in the areas of productivity, quality and communication (Yahya *et al.,* 2002:27; Fitzgerald *et al.,* 2002:3).

In view of that, methods are not perceived as relevant to practice, they are just being utilized because they appear plausible, hence the method-in-action (myriad of contextual factors pertaining to the process of development) has a great effect on the outcome of the development (Fitzgerald *et al.,* 2002:xii). Practically, systems development methodologies are frequently not fit for the development situation, and this can lead to the development of a method tailoring strategy to create the method-in-action to meet a particular situation (Fitzgerald *et al.,* 2002:100).

In such contexts, even with the notion of systems development methodologies, the underlying problems in the systems development have not been eliminated. One among other reasons for the failure has been that practitioners do not realize the benefits and needs of systems development methodologies; hence they are not in shape with the complexity of the development situations (Fitzgerald *et al.,* 2002:3).

2.2.5.4. Not widely practised

Systems development methodologies usage has been highly advertised in the academic research arena for many years, but it has not been universally embraced by business even if IT projects' failure is often experienced in the organizations (Griffin & Brandyberry, 2008:2; Mihailescu & Mihailescu, 2009:1). With such conceptualization, the evaluation of the reality of SDMs has not been keeping pace with their growth; therefore the veracity of their usefulness and effectiveness has not been accurately understood in the organizations (Siau & Tan, 2005:3132).

Practically, although systems development methodologies have long existed, they have not been noted for being terribly successful; and little has been known about them (Fowler, 2001:1).

2.2.5.5. Lack of the human factor

In particular, methodologies such as Structured Systems Analysis and Design Method (SSADM) have been claimed to be playing a crucial role to the success of systems development, however, they fail to note the human, social, and organizational aspects; instead, they emphasize the technical aspects of IS development**.** The deployment is not undertaken with people in mind, and this leads to users not being fully satisfied with the 'successful implementation' of the system (Yardley, 2002*:*83). And depressingly, Jackson Systems Development (JSD) was aimed at minimizing the responsibility of the developer in the development process; thus creating the illusion that the ability and insight of the developer are confined to the systems development methodology (Fitzgerald *et al.,* 2002:124).

2.2.5.6. Lack of full control

Another point to make is that some of the practices of systems development methodologies are carried out by external companies that specialize in providing information systems development services, it is alleged that the companies may lose control over the functions of their information systems and instead depend on technical influence and the breakthrough of external vendors (Laudon, 2000*:*384).

In summary, although the SDM can be assumed to have a significant impact on the successful completion of the IT projects (Dahiya & Jain, 2010), as per the 2009 Standish Group Report, more project failures and fewer project successes are still apparent. However, following an established methodology is much better than not following any methodology, as it will certainly be helpful in the planning and managing of some of the development key phases within the project (Yardley, 2002*:*83).

The next section presents and discusses different types of methodologies that are visible in the arena of information systems development.

2.2.6. Different types of Systems Development Methodologies (SDMs)

As we have learned from the aforementioned realities about SDMs there has been a great deal of interest in the systems development context over the years. Many new methodologies have been introduced in attempts to improve processes and developer acceptance (Griffin & Brandyberry, 2008:2). Different types of systems development methods require different techniques, sequences, steps and activities. However, it is assumed that all of these approaches may remain just presumptions if fail to achieve the purpose (Truex *et al.,* 2000*:*59).

This sub-section is aimed at making available knowledge about these types of SDMs in the development of information systems. The content is such that each systems development methodology is discussed according to the following components: philosophical approach, method, process model and tools and techniques.

2.2.6.1. Structured Systems Analysis and Design Method (SSADM)

- **Systems development method**

SSADM was developed by UK consultants Learmont and Burchett Management Systems (LBMS) and the Central Computing and Telecommunications Agency (CCTA) which are accountable to computer training and some forms of procurement in UK Civil Service (Middleton & McCollum, 2001:9; Avison & Fitzgerald, 2008*:*419; Weaver, 2004:67; Fitzgerald *et al*., 2002:33; Yardley, 2002:84). Initially, it was most popular in a number of government applications in UK (Yaghini *et al.,* 2009:5; Fitzgerald *et al*., 2002:33; Yardley, 2002:85). By now, it has changed dramatically in that results would accrue from technology's changes and methodology weak points (Yaghini *et al.,* 2009:5). Despite its name, SSADM is not generally structure-based but is instead regarded as a data-driven approach (Fitzgerald *et al*., 2002:33). In contrast, structured design is at its best for purposes of the selection and organization of programme modules to seek to solve the problems that emerge (Avison & Fitzgerald, 2008*:*395).

Furthermore, SSADM is commonly perceived as a 'burdensome' approach and it is not easy to implement. It is also viewed as a 'prescriptive approach' in that it prolongs the development period and these could lead to system delivery which does not comply with the initial specifications (Yardley, 2002:85). Other criticisms are that they are not easy to learn and be understood by staff and because of being too rigid, and not imitating the traditional practices of people (Middleton & McCollum, 2001:11).

In contrast, Weaver's (2004:67) contention is that SSADM is a very appropriate method when it comes to being a data-centric student project, regardless of the implementation environment; and with its nature of creating an unambiguous specification it can be mapped efficiently and effectively to virtually any programming environment. It is also claimed to be widely used and understood in the industry, thus it is grounded in best practice. The method is presumed to be comprehensive and most often used in medium to large projects, and produces deliverables to aid the progress at every step (Fitzgerald *et al*., 2002:33). Consequently, SSAM is most often used in organizations (Yardley, 2002:85).

- **Philosophical approach**

SSADM is classified as a weak (general) problem-solving approach in the information systems discipline (Meso *et al.,* 2006:18). It is believed to be a highly structured methodology and it emphasises data-modelling. SSADM provides strong guidance for its implementation and features comprehensive documentation which encompasses all facets of the information system project (Avison & Fitzgerald, 2008*:*419). On the other hand, Yaghini *et al. (*2009:5) state that SSADM is one of the blended methodologies where attention is on both data and process aspects in the information systems development process.

- **Process model**

SSADM is classified as a heavily 'prescriptive' methodology which allows some streamlining. Thus, it portrays a rational, engineering perspective of the world. It is believed to be data analysis-oriented in an emergent information system and it is driven by a 'waterfall' life cycle (Middleton & McCollum, 2001:10). The Waterfall cycle consists of a series of stages that will most often overlap and occur in sequence (Weaver, 2004:64).

SSDAM has seven stages (ranging from 0–6 in Figure 2.2.6.1) within a five-'module' framework. Each has its own set of plans, timescales, controls and monitoring procedures and links to the next module (Avison & Fitzgerald, 2008:420; Fitzgerald *et al*., 2002:33).

- **Feasibility study**
  0. *Feasibility*
     - Prepare for the feasibility study
     - Define the problem
     - Select the feasibility options
     - Create feasibility report
- **Requirements analysis**
  1. *Investigation of current environment*
     - Establish analysis framework
     - Investigate and define requirements
     - Investigate current processing
     - Investigate current data
     - Derive logical view of current services
     - Assemble investigation results
  2. *Business system options*
     - Define business system option
     - Select business system option
     - Define requirements
- **Requirements specification**
  3. *Definition of requirements*
     - Define requirements system processing
     - Develop required data model
     - Derive system functions
     - Enhance required data model
     - Develop specification prototypes
     - Develop processing specification
     - Confirm system objectives
     - Assemble requirements specification
- **Logical system specification**
  4. *Technical system options*
     - Define technical system options
     - Select technical system options
     - Define physical design module
  5. *Logical design*
     - Define user dialogues
     - Define update processes
     - Define enquiry process
     - Assemble logical design
- **Physical design**
  6. *Physical design*
     - Prepare for physical design
     - Create physical data design
     - Create function component implementation map
     - Optimize physical data design

Figure 2.2.6.1 SSADM by Avison and Fitzgerald (2008:420)

- **Tools and techniques**

It is presumed to be the most important methodology in this group. Its third edition introduced new techniques to contact users of the system in 1986. In version 4.0, a prototyping technique was mooted at first. Its last version, 4.3, was released in 1996 (Avison & Fitzgerald, 2008:419). CASE tools, Data Mate, LEAP, Prompt11, and computer tools serve as support to SSMD (Fitzgerald *et. al*., 2002:33).

2.2.6.2. Information Engineering Methodology (IEM)

- **Systems development method**

Information Engineering is a term that was first used by Clive Finkelstein to describe the data-modelling methodology that he developed in Australia in the late 1970s (Avison & Fitzgerald, 2008:434; Yaghini *et al.*, 2009:4).

Information engineering methodology is deemed to be an architectural approach that adheres to planning, analysing, designing, and implementing to enhance the management of its resources, including capital, people and information systems and in support of the accomplishment of business vision (Zwirowicz, 2009:8).

Information Engineering Methodology is a recent software methodology that enables a greater flexibility in the systems development process, and supports iterative operations (Meso *et al.*, 2006:18). It is known by its significant feature to promote communication among developers and users (Fitzgerald *et al*., 2002:31). Nonetheless, this methodology is still perceived to be rigid and have high levels of structure in the development process (Meso *et al.*, 2006:18).

- **Philosophical approach**

IE methodology focuses on the data aspect of information systems development and these data flows are assumed to be most significant in its operation (Leem & Kim, 2002:252; Yaghini *et al.,* 2009:4). It is believed to be a stable methodology from which to build information systems, hence, it successfully gives a positive approach to data-modelling. However, the IE approach still recognizes and considers the processes in the detail during the development information systems (Avison &

Fitzgerald, 2008:435). Furthermore, the IE approach is perceived to be a comprehensive methodology that addresses all aspects of life cycle. It can therefore be used for multiple purposes and in different ways in the development of information systems (Avison & Fitzgerald, 2008:435; Fitzgerald *et al*., 2002:33). Typically, IE methodology is a data-driven approach (Fitzgerald *et al*., 2002:33)

- **Process model**

Information Engineering supports iterative operations (Meso *et al.*, 2006:18). The principle of the iterative development is that development of the system occurs incrementally in a non-stop mode, and it follows the RAD approach (Fitzgerald *et al.,* 2002:38). Avison and Fitzgerald (2008:437) assumed IE methodology to be structured in a top-down tactic and it comprises the following four levels*:*

- Information strategy planning – the main objective here is to produce an information architecture and strategy to support overall objectives and needs of organization (Avison & Fitzgerald, 2008:437). It comprises four tasks, namely existing situation analysis, needs analysis, architecture definition and information strategic plan collection (Yaghini *et al.,* 2009:5; Avison & Fitzgerald, 2008:437).
- Business area analysis – the aim is to get an understanding of various business areas and establish their system requirements (Avison & Fitzgerald, 2008:437). It constitutes the following tasks: entity and functional analysis, interaction analysis, confirmation and planning for design (Yaghini *et al.,* 2009:5; Avison & Fitzgerald, 2008:437).
- Systems planning and design – the purpose of this level is to determine the performance of systems to fulfil the identified business requirements stipulated by the users (Avison & Fitzgerald, 2008:437). Concisely, systems designed from the collection of facts are linked to the technical factors in which they become the logical design (Avison & Fitzgerald, 2008:443). This level is divided into business system design and technical design (Yaghini *et al.*, 2009:5; Avison & Fitzgerald, 2008:443; Fitzgerald *et al.,* 2002:32) in which the former involves preliminary data structure design, system structure design, procedure design and confirmation with the latter earning attributes of

data design, software design, cutover design, operations design, verification of design, system test design and implementation planning (Avison & Fitzgerald, 2008:443).

- Construction and cutover – coding, testing and training of users are the main tasks undertaken in this phase, hence the implementation of the system to meet the previous levels demands is considered (Avison & Fitzgerald, 2008:437). Construction creates the defined implementation unit and it involves the following tasks: system generation and system verification. Alternatively, cutover is aimed at controlling the changeover from the current system and procedures to the new system and the tasks involved are preparation, installation of new software, final acceptance, and fan-out and system variant development. Succeeding the cutover, the running smooth operation of the system comprises the tasks that ensure that the system is maintained and that the changes in the system are carried out precisely. The tasks are to*:* evaluate the system, tuning and maintenance (Avison & Fitzgerald, 2008:445-446).

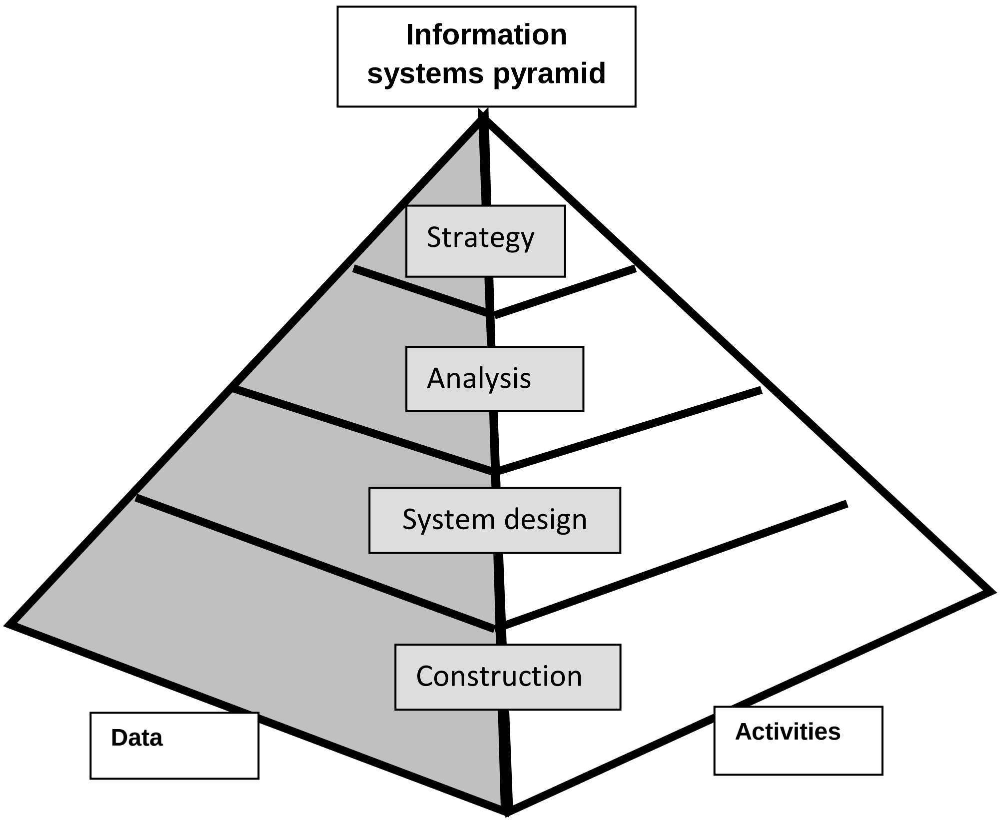


Figure 2.2.6.2: The four levels of IE (Avison & Fitzgerald, 2008:438)

The first two levels are technologically oriented, whereas levels three and four are technically oriented (Avison & Fitzgerald, 2008:437).

On the left of the pyramid are data; on the right are activities. Both the data and activities advance from a high-level, top management overview to a fully detailed implementation at the bottom (Avison & Fitzgerald, 2008:438).

- **Tools and techniques**

Entity Modelling, Normalization, Entity life-cycle, Decision tree, Decision tables, Structured English and Action diagrams are critical success factors. In the same way, it is also believed that the effective and efficient way of communication within IE methodology is through diagrams (Avison & Fitzgerald, 2008:436).

2.2.6.3. Rational Unified Process (RUP)

- **Systems development method**

The Rational Unified Process (UP) is an iterative software development process framework created by the Rational Software Corporation, a division of IBM since 2002 (Avison & Fitzgerald, 2008*:*461). It is an achievement of an integrated work of three primary methodologists, Ivar Jacobson, Grady Booch and James Rumbaugh (Yaghini *et al.*, 2009:6).

Brodie and Dalcher (2007*:*19) define the Rational Unified Model as a process expressed from the following three viewpoints:

- A dynamic perspective of the stages of the model;
- A statistic perspective that screens the process performance; and
- A practice perspective that proposes good practice.

Moreover, as per Fitzgerald *et al.* (2002:68), RUP is based on a set of six key principles:

  - Develop software iteratively;
  - Manage requirements;
  - Use component-based architectures;

- Model software visually;
- Verify software quality continuously; and
- Control changes to software.

Consequently, the advantage of RUP is that it can be modified to meet the emerged project needs and development context, and widely used by organizations which claim to have a successful record in systems development (Fitzgerald *et al.,* 2002:67).

- **Philosophical approach**

The Rational Unified Process (RUP) is an object-oriented methodology (Fitzgerald *et al.,* 2002:67; Yaghini *et al.,* 2009:5). This emphasizes the summation of data and process to define the object based on boundaries existing in the real world. The focus is on a thorough analysis of the 'what' part of the system rather than the 'how', holding off of the design phase (Yaghini *et al.,* 2009:5). The object-oriented approach gives rise to the encapsulation of data and the methodology attached to it in an object (Fitzgerald *et al.,* 2002:38).

- **Process model**

RUP is an iterative and incremental approach, and its requirements cannot be clearly defined at the initial point as they are prone to change and improve over time (Avison & Fitzgerald, 2008*:*460; Yaghini *et al.* 2009:6). The requirements are elicited through the use case, architecture-centric and iterative modes, and they usually utilize the Unified Model Language (UML) for modelling (Avison & Fitzgerald, 2008*:*460; Fitzgerald *et al.,* 2002:71).

Avison and Fitzgerald (2008, 460); Yaghini *et al.* (2009:6); Brodie and Dalcher (2007*:*19); Fitzgerald *et al.* (2002:67) point out that the Rational Unified Process (RUP) is a process model described from perspectives or phases of the model, an indication of process activities and the recommendation of good practices and it consists of four phases which are:

- Inception - establishes the business case by identifying external entities (people and system) that will interact with the emergent system (Brodie &

Dalcher, 2007:19, Avison & Fitzgerald, 2008:460). Feasibility in terms of cost and time estimation is assessed (Fitzgerald *et al.,* 2002:69).

- Elaboration – develops an understanding of the problem domain and the system architecture. It also includes the risks that may arise during systems development and therefore developing a requirement model and project development plan (Brodie & Dalcher, 2007:19; Avison & Fitzgerald, 2008:460; Fitzgerald *et al.,* 2002:69).
- Construction - includes design, code and testing of the system and pieces of the system can be comparably developed and thus be corporately ready to be delivered (Brodie & Dalcher, 2007:19; Avison & Fitzgerald, 2008:460; Fitzgerald *et al.,* 2002:70).
- Transition - requires that the system should be delivered to the respective users (Brodie & Dalcher, 2007:19; Fitzgerald *et al.,* 2002:70). Iteration is likely to occur in each of the phases and across the whole set of phases and the system is believed to be delivered in increments (Avison & Fitzgerald, 2008:460).

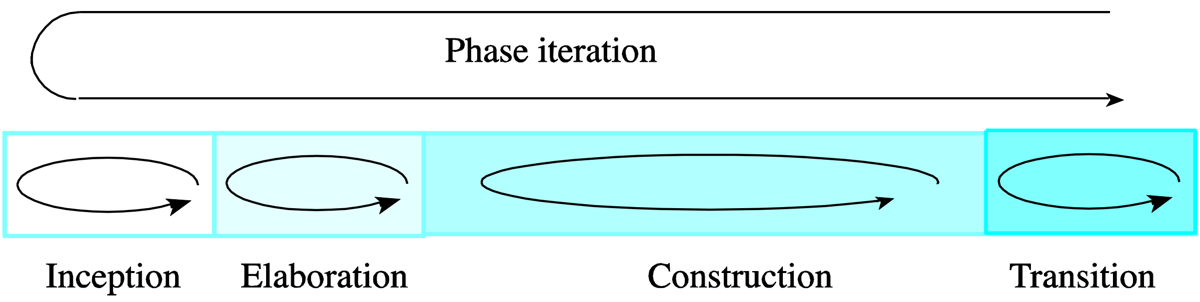


Figure 2.2.6.3a Rational Unified Process Model (Avison & Fitzgerald, 2008:460)

The RUP has nine process activities which are divided by core process workflows and core supporting workflows. The core process workflow consists of business modelling, requirements, analysis and design, implementation, testing and deployment (Avison & Fitzgerald, 2008:464; Fitzgerald *et al.,* 2002:70). Subsequently, core supporting processes consist of configuration and change management, project management and environment (Avison & Fitzgerald, 2008:464). These process activities are in conjunction with four phases.

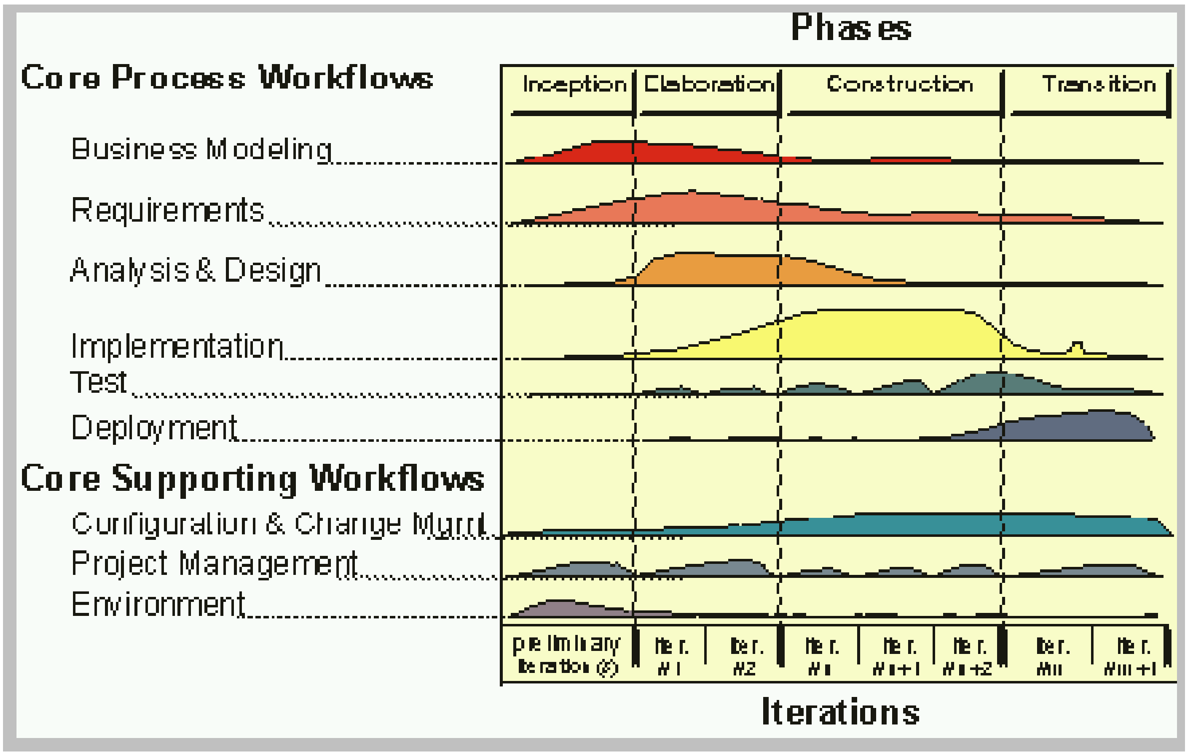


Figure 2.2.6.3b RUP process structure (Avison & Fitzgerald, 2008:464)

- **Tools and techniques**

Tools and techniques incorporated in RUP are*:* Object-oriented, UML (Unified Modelling Language), Class diagrams, Use-case diagrams, Interaction diagrams, CASE tool (Fitzgerald *et al.,* 2002:71-72), Sequence diagrams, state chart diagrams, Activity diagrams, Pert Chart and Gantt Charts.

2.2.6.4. Extreme Programming (XP)

- **Systems development method**

The Extreme Programming (XP) approach is believed to be a good practice that results in customer involvement to a greater degree (Sommerville, 2007*:*398). It is known to be one of the best methods in an agile approach which concerns the iterative development and persuades the system requirements that change rapidly during the IS development process (Sommerville, 2007*:*396). Moreover, it is more popular in its emphasis on communication and co-ordination among the team members at all times; and encourages teamwork among the stakeholders (customer, management and development team) to form the supportive business culture for the success of its implementation (Qumer & Henderson-Sellers, 2008:283; Avison & Fitzgerald, 2008:479; Hughes & Cotterell, 2009:279).

XP is more a series of principles for developing Information systems; it's unlike a step-by-step approach. It is mostly used in small or medium projects to develop software products of fewer than 10 and as a result, they are considered to be very quick in developing information systems, hence they are less detailed (Avison & Fitzgerald, 2008:479; Qumer & Henderson-Sellers, 2008:283).

Typically, XP is considered to have the following four core values, viz. simplicity, communication, feedback and encouragement (Hughes & Cotterell, 2009:95; Avison & Fitzgerald, 2008:479; Qumer & Henderson-Sellers, 2008:283). Therefore, it is assumed to be a 'lightweight methodology' (Fitzgerald *et al.,* 2002:61), in that it is not time-consuming as it does not have many complex rules that delay the progress of development (Avison & Fitzgerald, 2008:479; Qumer & Henderson-Sellers, 2008:283). Furthermore, it comprises the following phases, viz. planning, designing, and developing the code and productionalizing (Avison & Fitzgerald, 2008:480).

However, XP does also have some potential problems, in that its survival depends on tacit expertise and knowledge from which it plays contrary in terms of externalized knowledge in the formation of documentation. Its success depends on certain conditions which make its practices difficult when they are not implemented (Hughes & Cotterell, 2009:99).

- **Philosophical approach**

The philosophical approaches for XP methodology are rapid application development and an agile approach (Object-oriented), and the process models are incremental development, iterative development phases, hence it is claimed to be one of the popular agile methodologies (Sommerville, 2007:398; Qumer & Henderson-Sellers, 2008:283). The key features of agility are flexibility, speed, leanness, learning and responsibility (Qumer & Henderson-Sellers, 2008:282).

- **Process model**

It has been mentioned that the XP software development process has a great effect on iterative and rapid development (Sommerville, 2007*:*396). Thus, iteration development conviction is the ability of the system to deliver a succession of self-contained phases in which each phase resembles a stand-alone mini-project. Conversely, all the phases are grounded in similar sets of requirements and at a greater level of design (Weaver, 2004:59). Clearly, the application is broken down into small components and delivered in a sequence, and these components are inferred to be useful to the user (Hughes & Cotterell, 2009:88).

Consequently, the XP consists of the number of practices that support the principle of agile methods (Sommerville, 2007*:*400; Hughes & Cotterell, 2009:95; Fitzgerald *et al.,* 2002*:*61), and these principles involve:

  - Pair programming – two programmers work concurrently at one workstation, checking each other's work and providing support and encouragement to always do a good job.
  - Continuous integration – small changes are integrated most often (on a daily basis) in one base.
  - Collective ownership – every member has access to modifying development documents and coding at any given time.
  - Small releases – short release cycles are planned to be delivered.
  - Testing – unit tests (as classes expected to do by developers) and acceptance tests (as system is expected to do by users) are carried out continuously.
  - Refactoring – if a design does not work anymore, it needs to be thrown out.
  - Metaphor – system is based on metaphors that guide the developers throughout development.
  - Planning game – involves the requirements and project planning that break the release being planned down into iterations. The stories from users are broken down and tasks are summed up and compared to the iterations.
  - Coding conventions – all underlying source codes are labelled uniformly so that every team member can easily identify and understand them.

- Simple design – simple, easy and viable designs are used to meet users' immediate requirements.
- On-site customer – the user involvement is highly considered, and this helps the developers to always keep in focus the user's needs in order to develop systems that meet their goals.
- 40-hour week – if developers are working under pressure they are unlikely to be productive. Therefore, they need enough rest in order to work effectively and efficiently.

To a greater extent, the agility features (flexibility, speed, leanness, learning, responsiveness) become visible in the practices used in the phases of XP (Qumer & Henderson-Sellers, 2008:283).

Furthermore, XP suggests that software should be constantly made so that it gets improved and implemented instantly, hence the tendency for flexibility (to accept changes) and ease of understanding (as new stories are executed) are crucially important (Sommerville, 2007:401; Qumer & Henderson-Sellers, 2008:283).

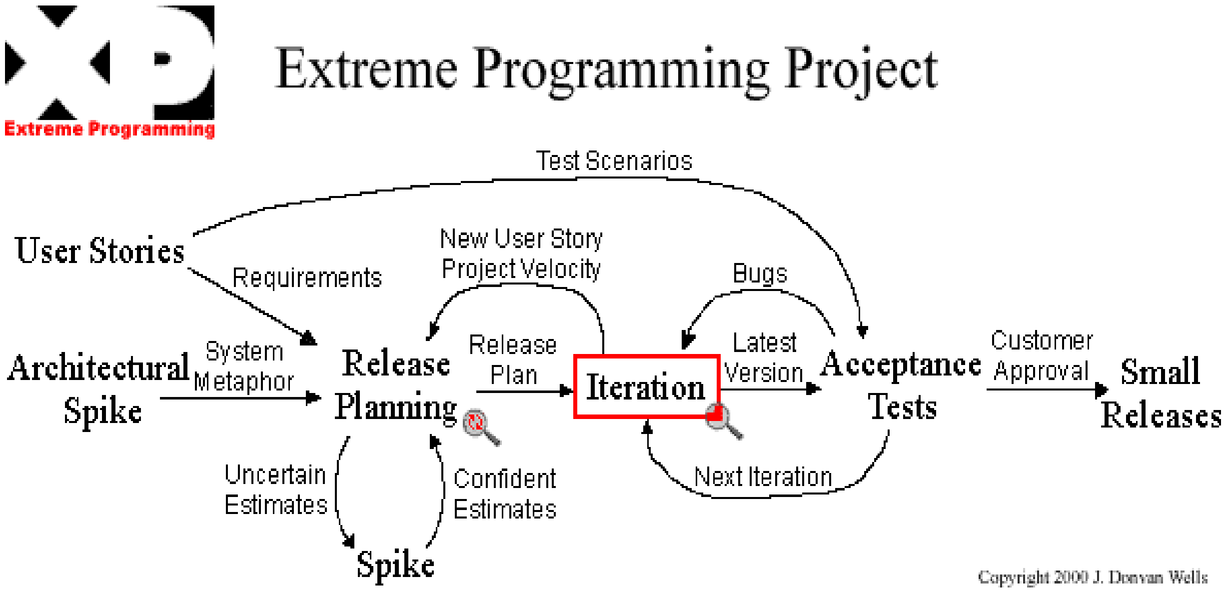


Figure 2.2.6.4 Extreme Programming

- **Tools and techniques**

XP uses users' stories to elicit the users' requirements (Avison & Fitzgerald, 2008:479; Sommerville, 2007:398; Huisman, 2009:16). The other techniques involve functional decomposition, time boxing, the pareto principle, prototyping (architectural spike), and users' stories and paired programming (Huisman, 2009*:*16).

2.2.6.5. Effective Technical and Human Implementation of Computer-based Systems Methodology (Ethics)

- **Systems development method**

ETHICS is a shortened way of representing Effective Technical and Human Implementation of Computer-based Systems. The name of this approach is premeditated to involve those methodologies that exemplify an ethical position. ETHICS is aimed at addressing the participative practices in the development of information systems (Yaghini *et al.,* 2009:6; Avison & Fitzgerald, 2008:487; Vidgen *et al.,* 2002:101). Yaghini *et al. (*2009:6) opinion is that, in order to develop an effective and efficient system, there must be a relation between the technology and technical aspects with social and organizational factors. Thus, technology must fit closely with social and organisational factors. Conversely, the other fundamental aim of this methodology is to develop a system of high-class quality work life and bestow an exclusive job satisfaction on the users (Avison & Fitzgerald, 2008:487).

Moreover, Vidgen *et al.* (2002:101) state that ETHICS is aimed at fostering genuine participation, walking the extra mile in providing input to requirements and specifications and doing the prototype evaluation afterwards. In a simple way, it emphasises the in-cooperation of human interest during the development of the information system. Even though, Avison and Fitzgerald (2008:489) argue that the ETHICS methodology has a fairly high level of freedom and flexibility, one of the criticisms is that it can be used for managerialist purposes, to impose on the workers to accept declined changes at the expense of user involvement to gain increased efficiency (Vidgen *et al.,* 2002:109).

- **Philosophical approach**

The first philosophy of ETHICS, which is mostly at a distance from other methodologies, is that the organizational conduct is with the perception that the development of computer systems should not only engage the technical issues but consider the organizational issues as they are of utmost significance in the development of systems. These beliefs are judged to be a core concern grounded on the socio-technical approach (Avison & Fitzgerald, 2008:487). According to Vidgen *et al.* (2002:33), the socio-technical approach in the information system incurs job satisfaction and user participation during the development process. Job satisfaction is a goodness of fit between the expectations and the needs of employees from work and what they are obliged to do as required by the organizations (Vidgen *et al.,* 2002:103). Five areas of measurements (Table 2.2.6.5) needed to ascertain the goodness of the fit are identified, namely the knowledge fit, the psychological fit, the efficiency fit, the task structure fit and the ethical fit (Avison & Fitzgerald, 2008:487-488; Vidgen *et al.,* 2002:104-105).

Fit	**Employees' needs**
KNOWLEDGE	Want personal skills and knowledge to be used and developed
PSYCHOLOGICAL	Seek to further personal interests, e.g., sense of achievement, recognition, advancement, status
EFFICIENCY	Seek equitable effort-reward bargain and acceptable supervisory controls. Seek efficient support services such as information and technology.
TASK STRUCTURE	Seek a set of tasks with variety, interest, targets, feedback, task identity, and autonomy
ETHICAL	Seek to work for an employer whose values do not contravene personal values

Table 2.2.6.5: ETHICS Fit for job satisfaction (adapted from Vidgen *et al.,* 2002:103)

The second philosophical view of ETHICS methodology is user participation. This involves all the affected people in the emergent system, namely direct and indirect users, top managers, team managers, suppliers, customers, developers, and so on (Avison & Fitzgerald, 2008:488). Participation usually includes the setting up of steering committees and a design group or groups. The task of a steering committee involves the setting up of guidelines for the design group, and this leads to the evolution of the new system that constitutes choice of hardware and software, human-computer interaction, workplace reorganization and allocation of responsibility (Avison & Fitzgerald, 2008:490).

- **Process model**

ETHICS methodology comprises the following 15 steps (Avison & Fitzgerald, 2008:491):

1) Why change?;
2) system boundaries;
3) description of existing system;
4), 5), and 6) definition of key objectives and tasks;
7) diagnosis of efficiency needs;
8) diagnosis of job satisfaction needs;
9) future analysis;
10) specifying and weighing efficiency and job satisfaction needs;
11) the organisational design of the new system;
12) technical options;
13) the preparation of a detailed work design;
14) implementation;, and
15) evaluation.

The importance of user involvement and participation obtained by ETHICS methodology has sustained the process of change that takes place over time as compared to all information systems development methodologies (Avison & Fitzgerald, 2008:496).

- **Tools and techniques**

To ensure the participation of those affected by the system, role player analysis, JAD (Joint Application Development) techniques are deemed to be suitable. Hence, JAD is considered to be the best in situations where high participation is necessary (Avison & Fitzgerald, 2008:131). The other way to verify the quality of job satisfaction is the use of a Questionnaire (Vidgen *et al.*, 2002:107). WebQual tools investigate how users interact with websites, whereas a quality functional development is used to elicit the user's opinions through every stage of project development (Vidgen *et al.,* 2002:112).

2.2.6.6. Soft Systems Methodology (SSM)

- **Systems development method**

Soft systems as methodology (SSM) was developed by Peter Checkland at the University of Lancaster in the 1970s as an antidote to ‘hard’ systems thinking as exemplified by the work of the Rand Corporation in the 1960s (Vidgen *et al.,* 2002:80; Yaghini *et al.*, 2009:7).

It is an approach behind systems thinking (effective in a situation where they are perceived as a mess, where there is a need to define a problem even before trying to solve the problem) and they are contrasted with the reductionist approach (consider the whole part and break it down in order to gain an understanding of traditional analysis (Vidgen *et al.,* 2002:63). Sensibly, a soft systems methodology seeks to understand the problem rather than rushing in to try to solve it.

A soft systems methodology places an emphasis on human activity systems, i.e. people are involved in the activity within an organisation. The methodology paves the way in which the complications of such human interaction can be investigated, described and ultimately understood (Warwick, 2008:272). Understanding of the situation analysis provokes the methodology to distinguish change that is both systemically desirable (in that it will alleviate some of the problems and issues) and culturally feasible (actors within the system will be inclined to engage with the changes proposed and the change process by itself (Warwick, 2008:272; Vidgen *et al.,* 2002:63)).

As a consequence, there is much emphasis on SSM to encourage learning and the promotion of shared understanding that would serve as a change to influence problem resolutions (Warwick, 2008:272; Vidgen *et al.,* 2002:95).

Although the optimistic picture about the soft system methodologies has been presented, according to Vidgen *et al.* (2002:13) SSMs have been criticized for their failure to address critical aspects of organizational changes in terms of the power of vested interests. In addition, soft system methodologies concentrate much less on tools and techniques than most of the other ‘hard’ methodologies (Avison & Fitzgerald, 2006:514).

- **Philosophical approach**

SSM is judged to be a system-based methodology in which the system rules and principles that allow the structuring of thinking about the real world are exploited. The focal point of SSM is on the soft (human) aspect of the system on the assumption that the people in the organization are affected to a greater degree by the information systems development (Yaghini *et al.,* 2009:6).

Interestingly, SSM constitutes the main modes, namely the real world activities and the systems thinking about the real world (Vidgen *et al.,* 2002:82), whereas the real world activities involve interviews and meetings to gain an understanding of the problem situation from which the theory is represented by the use of 'rich pictures’. On the other hand, a systems thinking approach involves the concept of hierarchical structure and entails the inclusion of conceptual model, communication, control, and emergent properties to identify 'relevant systems' which may reveal the useful insight (Warwick, 2008:287; Vidgen *et al.,* 2002:80). Consequently, the simplicity of thinking towards modelling process lies in its evidence in comparing the real systems that have progressed over time and have become visible (Warwick, 2008:287).

Soft systems methodologies have adapted the lifestyle of scientists and engineers where they have traditionally been raised on the principle of reductionism so that analysis of a problem is based on the structure and decomposition to reveal how things work (Warwick, 2008:272).

- **Process model**

Warwick (2008:19) and Vidgen *et al.* (2002:82) indicate that soft systems methodologies are often described as processes of investigation and learning and is most popularly presented as a seven-stage model as shown in Figure 2.2.6.6. This process model emerged during the process of evolution, and these phases underpin the iterative development.

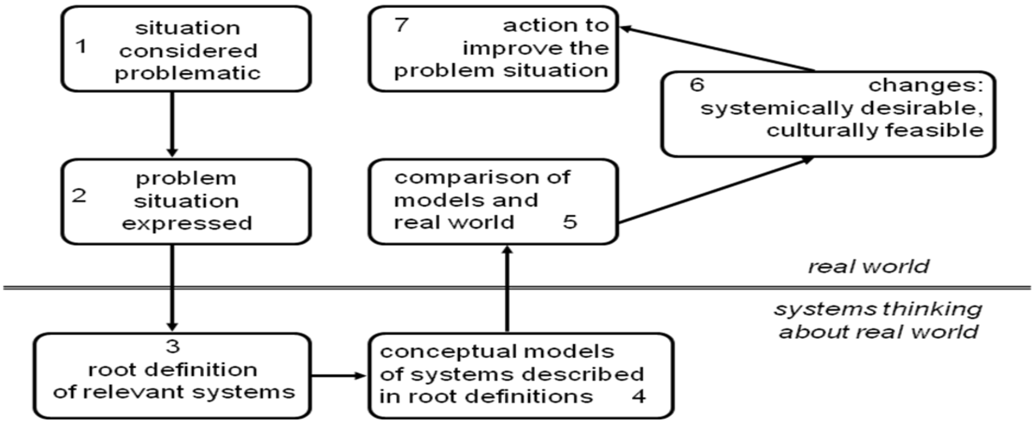


Figure 2.2.6.6 Seven-stage model of SSM

- **Tools and techniques**

To make sense of the concept of SSM, the root definitions, conceptual model and rich picture are utilized to capture the views concerning the situation (Vidgen *et al.,* 2002:86). The root definitions comply with the defining features of a system, the transformation which defines the purpose of the system. The construction of the conceptual model is evolved from the root definitions, and this allows the system to be viewed from a different perspective and also to describe the system process in order to accomplish the desired transformation. Consequently, root definitions and conceptual models have been derived systematically to form rich pictures which give a pictorial description of the scenario and offer a focus to pursue the discussion and analysis (Warwick, 2008:287).

The well-known mnemonic CATWOEA is the concept used to illustrate the element of the root definition (Warwick, 2008:278). Vidgen *et al.* (2002:13) argue that soft systems methodologies use the mnemonic "CATWOE" (customers, actors, and transformation process, worldview, and owner, environmental constraints) to list the perspectives of a situation that must be moderated (Table 2.2.6.6).

Customer	The victims or beneficiaries of the transformation
Actor	Those who would do the transformation
Transformation	The conversion of input to output
Weltanschauung or worldview	The worldview which makes the transformation meaningful
Owner	Those who could stop the transformation
Environmental constraints	Elements outside the system which it takes as given

Table 2.2.6.6: CATWOE mnemonic (adapted from Vidgen *et al.,* 2002:89)

2.2.6.7. The Yourdon systems Method (YSM)

- **Systems development method**

The Yourdon systems method (YSM) is similar to STRADIS, and it applies functional decomposition where the problem is broken down into manageable units for its success (Avison & Fitzgerald, 2008:402). The reason behind the innovation of Yourdon systems method (YSM) was to eliminate inexpressiveness which is very apparent in process-oriented methods (Yaghini *et al.,* 2009:5). It covers every practice that occurred in the organization and in the system itself (Avison & Fitzgerald, 2008:402). The Yourdon method is one of the structured approaches that represent the 'hard' development method.

Though structured approaches have claimed to be popular in the organizations, they have always been criticized as being too time-consuming in their implementation and that the practitioners are reluctant to their acceptance (Fitzgerald *et al.,* 2002:29).

- **Philosophical approach**

The Yourdon systems method (YSM) is a process-oriented methodology which applies the existing processes to information systems development (Yaghini *et al.,* 2009:5). Though it actuates the process, it has a sense of data analysis as compared to STRADIS. This data process triggers 'middle-up' approach called event-partitioning rather than a top-down approach (functional decomposition) (Avison & Fitzgerald, 2008:390).

- **Process model**

The Yourdon systems method consists of three phases, with each of them constituting some underlying steps. Its steps and phases are as follows (Yaghini *et al.,* 2009:5; Avison & Fitzgerald, 2008:402):

- Phase 1*:* Feasibility - Interviewing users, Building DFD, Building ERD (Yaghini *et al.,* 2009:5), studying the current system (Avison & Fitzgerald, 2008:403).
- Phase 2*:* Main model construction - Building Enterprise Essential Model, System Essential Model (Yaghini *et al.,* 2009:5). Statement of purpose, content diagram and event list (Avison & Fitzgerald, 2008:403).
- Phase 3*:* Performance model-building - Building Enterprise Performance Model and System Performance Model (Yaghini *et al.*, 2009:5). Model software environment in which the system is to exist, model structure of the software to be produced (Avison & Fitzgerald, 2008:403).

- **Tools and techniques**

Yourdon systems methods (YSM) underpin the following behavioural model, viz. data-flow diagram, entity relationship diagram, state transition diagram, data dictionary and process specifications (Avison & Fitzgerald, 2008:403).

2.2.6.8. Jackson Systems development (JSD)

- **Systems development method**

The Jackson systems development method has been developed from Jackson structured development (JSP) in which it has more of an effect on programming practices (Fitzgerald, *et al.,* 2002:34; Avison & Fitzgerald, 2008:407). The JSD is one of the structured approaches underpinning the hard end of the spectrum (Fitzgerald, 2002:34). Jackson discovered the following traits pertinent to JSP, namely non-inspirational and teachable, as such, substantiate the similar results from different programmers in a particular developing situation; practicality and ease to understand, thus confirming access to the novice programmer (Fitzgerald *et al.,* 2002:34). Furthermore, Jackson structured development is deemed to be grounded by reason and rational principles, and therefore believed to be concerned with the modelling of the real-world problem situation (Avison & Fitzgerald, 2008:408; Fitzgerald, *et al.,* 2002:34).

Jackson structured development with its high sense of reliance on intuition and creativity on the programmer side is claimed to be widely used in organisations (Fitzgerald *et al.,* 2002:34). However, it is criticised for its emphasis on software development practices instead of addressing the organisational needs and also its failure to attend to all phases of the life cycle development (Avison & Fitzgerald, 2008:407).

- **Philosophical approach**

The notion of functional decomposition which is common in structural development was regarded as the limitation in the invention of the Jackson structured development, and this perception that was uncovered by Jackson further has an effect on method for the greater reliance on intuition and creativity in the favour of the programmer (Fitzgerald *et al.* 2002:34). In contrast, the Jackson structured development was aimed at reducing the insight that could be gained by the programmer during the development of information systems, and claimed to underpin a composition process (Fitzgerald *et al.,* 2002:34). As such, is regarded as data-driven oriented unlike the structured approach (Fitzgerald *et al.,* 2002:34).

- **Process model**

The JSD method entails three main phases, viz. the model phase (concerns modelling the actions and entities and their attributes), the network phase (construction of specifications from the models), and the implementation phase (deals with running the processes and storing the data). These phases constitute the following steps (Avison & Fitzgerald, 2008:408; Fitzgerald, *et al.,* 2002:36):

    - Modelling phase*:* Entity/Action Step: Entities and actions are identified to define the real world aspects that would affect the system (Avison & Fitzgerald, 2008:410).

- Entity Structure Step: Entities and actions appear in a diagram shape with regard to the sequence, selection and iteration (Avison & Fitzgerald, 2008:410).
- Network phase: Initial Model Step - Creation of a model to symbolize the real world (Avison & Fitzgerald, 2008:410).
- Function Step: Addition of functions to the model created in the initial model to produce the outcomes that would occur when a combination of events is taking place (Avison & Fitzgerald, 2008:414).
- System Timing: Gathering of delays information that takes place in and between the process amongst input and output, in which this information informs the decisions (Avison & Fitzgerald, 2008:416).
- Implementation Step: Physical system specification step: Concerns the sharing of processors among processes (Avison & Fitzgerald, 2008:416), thus ensuring system efficiency (Fitzgerald, 2002:36).

- **Tools and techniques**

The JSD is envisaged as using the following tools and techniques, namely a.) the structure diagram for the life time span of entities and actions (Avison & Fitzgerald, 2008:410), b.) Pseudo-code to define each modelling process (Avison & Fitzgerald, 2008:411), c.) System-specification diagram to express the communication between processes, d.) state vector ensures the connection between the button and process (Avison & Fitzgerald, 2008:413).

2.2.6.9. Web Information Systems Development Methodology (WISDM)

- **Systems development method**

A Web Information Systems Development Methodology (WISDM) is an information systems development methodology that is purposely designed to develop the web information system (Venable & Lim, 2001:1). In many attempts to develop web-based information system, the focus has been on the user interface, in particular the look and feel of a website; however, they have failed to address wider aspects of information systems (Vidgen *et al.,* 2002:29; Avison & Fitzgerald, 2008:481). WISDM is articulated through the Multiview framework that emphasises the significance of technical artefacts, organisational activities and individual mutual in IS development (Vidgen *et al.,* 2002:30). Certainly, the WISDM traditional view is to incorporate the socio-technical (social and technical aspects) approach in IS development (Avison & Fitzgerald, 2008*:*483).

Out of all types of WISDM, namely OOHDM, Dexter Hypertext Reference Model, NLG approach, Web development tool methodology, none was ever familiar except RMM and W3DT which were claimed to be the ones tried and tested. Instead, some form of methodology was used to guide the web information systems development, in some cases, no methodology was ever used. This authenticates the perception that none of the methods proposed to develop web applications have been well-thought out across the board (Avison & Fitzgerald, 2008*:*481).

Alternatively, the Multiview framework for generating WISDM is at its best when advocating a high sense of interaction between IS developer (and other change agents) and the problem situation (Vidgen *et al.,* 2002:30).

- **Philosophical approach**

There has been a strong thrust among traditional information systems development methodologies from the waterfall life-cycle to rapid application (RAD) to adapt web-specific aspects to their techniques and practices (Vidgen *et al.,* 2002:29; Avison & Fitzgerald, 2008*:*481). Hence the Multiview framework approach of WISDM was introduced and its primary aim is to serve as a basis for web information systems development methodology (WISDM) (Vidgen *et al.,* 2002:29). The fundamental assumption is that the multiview framework is purported to provide the fundamentals to develop the situation-specific methodology, thus considering the social and contextual aspects of making sense of IS development. Essentially, the philosophical views involve the functionalist paradigm of objectivism and social order (Vidgen *et al.,* 2002:30). The multiview framework is characterized in the following, viz. general framework, local, emergent methodologies (Vidgen *et al.,* 2002:31).

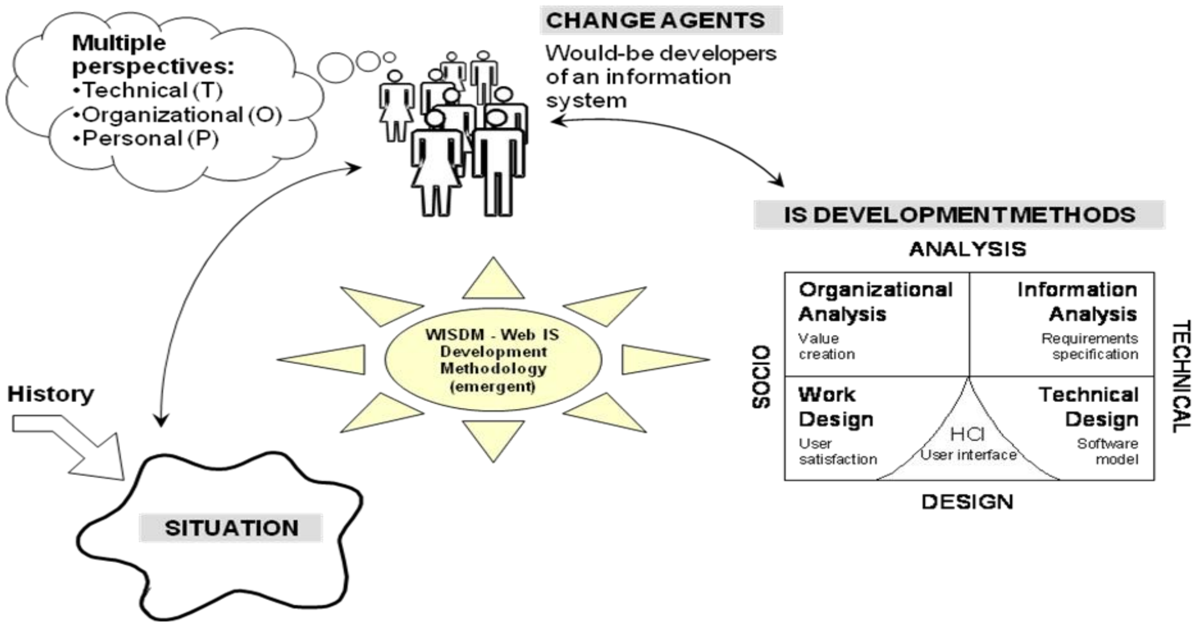


Figure 2.2.6.9.a The multiview framework for generating a WISDM (Vidgen *et al.,* 2002:29)

- **Process model**

Within the Multiview framework, there are methods that the developer (change agents) can draw on to interact with the problem situation with the aim of creating the methodology (WISDM) in practice. These information system methods are reviewed in a matrix (Figure 2.2.6.9b) (Vidgen *et al.,* 2002:32). A methods matrix is characterized by five components: organizational analysis, work design, information analysis, technical design and human computer interaction design. Therefore, there is no a priori ordering of the five aspects of the WISDM matrix as in conventional methodologies, thus each category becomes vigorous as is appropriately required during the execution of the project (Avison & Fitzgerald, 2006:483).

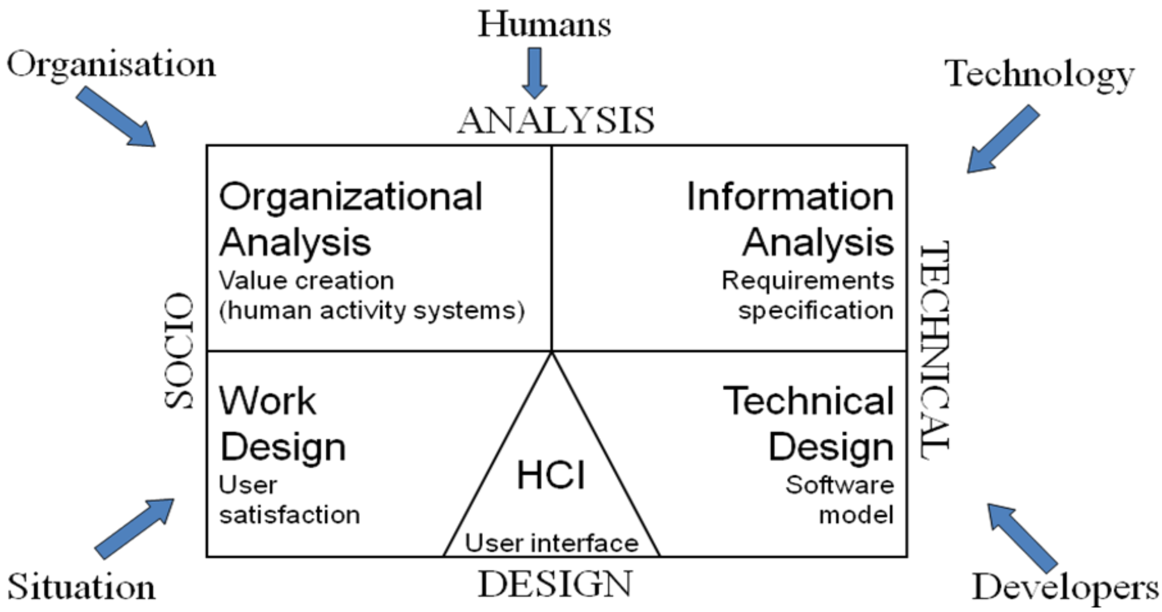


Figure 2.2.6.9b WISD Matrix

- **Tools and techniques**

The approach in WISDM is aimed at using the UML from which the structural (class model), behavioural (user-case and interaction diagrams) and process-oriented (activity diagrams) models of the problem situation are constructed (Avison & Fitzgerald, 2006:483).

2.2.6.10. Structured Analysis Design and Implementation of information Systems Methodology (STRADIS)

In practice, Structured Analysis Design and Implementation of information Systems Development Methodology were developed by Chris Gane and Trish Sarson in 1979 (Avison & Fitzgerald, 2008:395; Kimble, 2008; Kotsonis, 2007:10). The structured approach to design was first proposed by Myers and Constantine (1974), which in turn was refined by Yourdon and Constantine (1978) and the Jackson Systems development (Avison & Fitzgerald, 2008:395). It is claimed to be the methodology that can be used for the development of both small and large systems as well as for automated or non-automated systems (Avison & Fitzgerald, 2008:396). Furthermore, it is concerned with the selection and organisation of programme modules and interfaces and can be used when there is a backlog of systems waiting to be developed or insufficient resources available to develop all of the potential systems (Kimble, 2008)

- **Philosophical approach**

STRADIS is based on process modelling and reflects the functional decomposition and the use of the techniques such as DFD, Decision trees, Decision tables, Structure English (Avison & Fitzgerald, 2008:396; Kimble, 2008; Kotsonis, 2007:22).

- **Process model**

STRADIS consists of phases which focus on analysis, less on design, and views implementation and testing as a linear process (Kotsonis, 2007:10). The phases are as follows:

- Initial study (Avison & Fitzgerald, 2008:396; Kimble, 2008)

The purpose of this phase is to ensure that the fundamental aspects of a system are developed. The main criteria in this process are the monetary costs and benefits of each proposal.

- Detailed study (Avison & Fitzgerald, 2008:396; Kimble, 2008)

Upon the approval of the initial study, further potential users and all stakeholders of the system are identified, namely top managers become responsible for profit, middle managers are expected to take charge and end-users will work with the developers to establish system requirements.

- Defining and designing alternative solutions (Avison & Fitzgerald, 2008:396; Kimble, 2008)

System analysts and developers work together to establish three alternative designs together with estimates of costs, benefits, time-scales, hardware and software

  - A low-budget, quick implementation which may not meet all the objectives.
  - A mid-budget, medium-term version, which achieves a majority of the objectives.

    - A higher budget, more ambitious version achieving all the objectives.

- Physical Design (Avison & Fitzgerald, 2008:396; Kimble, 2008)

The designer seeks to identify either of two alternative structures for the system:

    - All transactions follow very similar processing paths – this is termed a ‘transform-centred’ system.
    - Transactions require different processing - this is termed a ‘transaction-centred’ system.

- **Tools and techniques**

STRADIS employs a combination of techniques such as DFD, decision trees, decision tables, structured English, data dictionary and a data structure diagram (Avison & Fitzgerald, 2008:242-272; Kimble, 2008).

In summary, the above different types of systems development methodologies have been categorized in different types of classifications, namely the process-oriented approach which reflects the YSM and JSP; the blended-oriented approach which exemplifies SSDM; the data-oriented approach which has a strong effect on IE; the organizational-oriented approach which constitutes the SSM; and the people-oriented approach which comprises the ETHICS. Interestingly, the WISD is reflected within the Multiview framework (Avison & Fitzgerald, 2008:390).

The subsequent sub-section provides the framework which compares the above system methodologies based on the key features of each methodology. However, XP and WISDM are not compared to other SDMs due to their exceptional natures.

2.2.7. Framework for comparing methodologies

This sub-section is aimed at presenting a comparison of the methodologies discussed above. The comparison is depicted through the framework shown in Table 2.2.7. The comparison is based on the following elements (some of the elements are broken into sub-elements): philosophy (paradigm, objectives, domain and target), model, tools, scope and practice (background and participants). The framework is constructed as per observations by Yaghini *et al.* (2009:9-10) and Avison and Fitzgerald (2008:604-613).

The signs are used to represent the following:

■ Methodology covers the stage with some details.

□ Methodology addresses the area but with fewer details and less depth.

☼ It is introduced in methodology but no procedure, technique, or rule is provided.

Practice	Participants	Professional IS	Professional IS	Professional IS	Professional IS	System users & expert	System users & expert	Professional IS	Professional IS
	Back-ground	Commercial	Commercial	Commercial	Academic	Academic	Academic	Commercial	Academic
Scope	Maintenance	☼							
	Evaluation	☼		□	□	□			
	Implementation		□	□	■	□			☼
	Testing	□		■	■				□
	Programming	□		□	□				■
	Physical design	■	■	■	■	□		■	■
	Logical design	■	■	■	■	■		■	■
	Analysis	■	■	■	■	■	■	■	□
	Feasibility	■	■	□	□	□	□	■	
	Strategy		□	■	☼		■	☼	
Tools		Use to a great extent	Advise to use	Complexity and time	Complexity and time	-	-	Advise to use	Advise to use
Model		DFD	DFD	ERD	Object Model	Process oriented	Process oriented	DFD	Process oriented

SDM	Approach	Philosophy			
		Paradigm	Objectives	Domain	Target
STRADIS	Hard	Scientific	Computerized IS building	Problem solving method	Any size of system
SSADM	Hard	Scientific	Computerized IS building	Problem solving method	Big system
IE	Hard	Scientific	Computerized IS building	IS organizational need	Big system
RUP	Hard	Scientific	Computerized IS building	Problem solving method	Big system
ETHICS	Soft	Systematic	Complex situation	Problem solving method	Any size of system
SSM	Soft	Systematic	Improvements	IS organizational need	Human activities
YSM	Hard	Scientific	Computerized IS building	Problem solving method	Big system
JSD	Hard	Scientific	Computerized IS building	Problem solving method	Real time application

Table 2.2.7: Framework for comparing different types of SDMs

Section Two

2.3. A discussion of critical success factors (CSFs)

2.3.1. Background

The purpose of this section is to present a literature review of the critical success factors (CSFs) which affect the success of information technology (IT) projects in organizations. Based on the investigated views from the authors, the section establishes an insight into the most significant critical success factors likely to have an impact on information technology (IT) project success.

The previous researchers have proved that companies have difficulty with information technology (IT) projects to be completed on time or on budget. Hence, many IT projects are cancelled before completion or are not even implemented (Al Neimat, 2005:3). The concept behind critical success factors (CSFs) is that in every organisation there are selective functions and practices in the sense that, if they can be pursued appropriately, the organisation will perform well in terms of the development of IT projects. As such, organizations must at all times measure their performance against these practices, and where necessary they should take corrective action (Wetherbe, 2001:530).

To give a sense of background, Dalcher and Brodie (2007:273) maintain that a success factor is believed to be an endeavour aimed at predicting the good practices needed to accomplish the project successfully. Though the lack of success factors does not ensure a failure, yet it can be perceived as a phenomenon that can jeopardize the project, and as a result, the project will probably fail to reach the desired results.

Practically, the project success is normally thought of as the achievement of some pre-determined project goals, such as time, cost, and scope (Schwalbe, 2007:13). Dalcher and Brodie (2007:8) further emphasise the notion that for a project to be a success, it must meet its requirements and be completed such that it is:

- Within budget
- Within time
- At the desired performance level
- With acceptable quality
- Offering at least the minimum agreed functionality
- Utilizing the assigned resources effectively and efficiently
- Accepted by the clients
- Used by the intended users
- Delivering the promised benefits (which should ideally exceed the costs)

Moreover, The Standish Group (1994:2) categorizes projects into three resolution types:

- Project success: The project is completed on-time and on-budget, with all features and functions within the scope.
- Project challenged: The project is completed and operational but over-budget, over the time estimate, and offers fewer features and functions than initially specified and planned.
- Project impaired: The project is cancelled at some point during the development cycle or never implemented.

Subsequently, the section will identify the top ten most critical success factors as cited in the literature.

For purposes of gaining more knowledge on the issue of failures and successes of information technology (IT) projects, the literature will examine the subsequent Standish Group Chaos Reports. The Standish Group was from 1994 to 2009 dedicated to analysing how information technology (IT) projects were managed and performed in companies.

2.3.2. The list of critical success factors (CSFs) from the literature study

This section outlines the critical success factors and collates them according to previously related research outcomes. Table 2.3.2 lists the critical success factors (CSFs) per order of importance as presented from the previous similar literature survey. The table entails the sub-critical success factor, literature citations, domains and frequency of citations.

Critical success factor	Sub-critical success factor	Literature citation	Domain	Frequency of citation
Top management involvement and support	Top Management involvement	Al Neimat (2005:3)	IT Projects	32
		Al-Mudimigh *et al.* (2010:5)	IT Projects	
		Arntzen and Nkosi (2009:215)	Knowledge Based Systems	
		Belout and Gauvreau (2004:5)	Project Management	
		Biehl (2007:55)	Information Systems	
		Blindenbach-Driessen, and van den Ende (2006:557)	Project-based firms	
		Chaos Reports (2009)	IT Projects	
		Ehie and Madsen (2005:553)	ERP	
		Jarrar *et al.* (2010:7)	ERP	
	Leadership	Johnson (2000:70)	IT Projects	
		Kronbichler *et al.* (2009:15)	ERP	

		Lu *et al.* (2008*:*359)	Information Systems	
		Moohebat *et al.* (2010:105)	ERP	
		Nasir and Sahibuddin (2011*:*2182)	Software projects	
		Ngai *et al.* (2008:1)	ERP	
		Olson and Zhao (2007:136)	ERP	
	Committed and informed executive sponsor	Poon and Wagner (2001*:*395)	Information systems	
		Rainer and Hall (2002*:*81)	Software projects	
		Remus (2006*:*6)	Portal implementation	
		Sun *et al.* (2005:193)	ERP	
		Suxia Gai and Congwei Xu (2009*:*563)	Knowledge Based Systems	
		Tan *et al.* (2009*:*6)	Information Systems	
	Executive Support	The Standish Group (1995:5)	IT Projects	
		The Standish Group (1995:5)	IT Projects	
		Wixoma and Watson (2001*:*33)	Management Information Systems	
		Wong (2005*:*266)	Knowledge Based Systems	
		Xulong *et al.* (2008*:*4)	Information Systems	
		Yahya *et al.* (2002:29)	Information Systems	
		Yardley (2002:115)	IT Projects	
		Yeo (2002*:*245)	Information Systems	
		Yingjie (2005:44)	ERP	
		Young and Jordan (2008:719)	Project Management	
Team composition & teamwork	Team work	Kronbichler *et al.* (2009*:*15)	ERP	29
		Ngai *et al.* (2008:1)	ERP	
		Sun *et al.* (2005:193)	ERP	

		Al Neimat (2005*:*3)	IT Projects	
		Arntzen and Nkosi (2009*:*215)	Knowledge Based Systems	
		Chaos Reports (2009)	IT Projects	
		Chow and Cao (2008*:*970)	Software projects	
		Dalcher and Brodie (2007: 8)	Software projects	
		Humphrey (2005*:*28)	Software Project	
		Nah *et a*l. (2001*:*289)	ERP	
		Remus (2006*:*6)	Portal implementation	
		Tesch *et al.* (2009:663)	Information systems	
		Wixoma and Watso (2001*:*33)	Management Information Systems	
		Yahya *et al*. (2002:29)	Information Systems	
		Yardley (2002:117)	IT Projects	
	Staff training	Biehl (2007*:*55)	Information Systems	
		Gai and Xu (2009*:*563)	Software Project	
		Jarrar. *et al.* (2010*:*7)	ERP	
		Olson and Zhao (2007:136)	ERP	
		Rainer and Hall (2002*:*81)	Software projects	
		Schwalbe (2007*:*17)	IT Projects	
		Yingjie (2005:44)	ERP	
	Sufficient Staff	Lind and Culler (2009:6)	IT Projects	
		Belout and Gauvreau (2004:5)	Project Management	
		Young and Jordan (2008:719)	Project Management	
	Motivate the team	Nasir and Sahibuddin (2011*:*2182)	IT Projects	
		Wong (2005*:*267)	Software projects	
		Wourms (2004*:*2)	Knowledge- based	

			Systems	
	Appropriate IS Staff	Poon and Wagner (2001:395)	Information systems	
Project Management	Project management practices	Chow and Cao (2008:970)	Software projects	27
		Ehie and Madsen (2005:553)	ERP	
		Kronbichler *et al.* (2009:15)	ERP	
		Moohebat *et al.* (2010:105)	ERP	
		Nasir and Sahibuddin (2011:2182)	Software projects	
		Ngai *et al.* (2008:1)	ERP	
		Olson and Zhao (2007:136)	ERP	
		Remus (2006:6)	Portal implementation	
		Tan *et al.* (2009:7)	Information Systems	
		Yardley (2002:55)	IT Projects	
		Yingjie (2005:44)	ERP	
		Young and Jordan (2008:719)	Project Management	
	Deliverable dates / smaller scope	Al Neimat (2005:3)	IT Projects	
		Benediktsson (2002:7)	Software Projects	
		Yeo (2002:245)	Information Systems	
	Identify and manage complexity/risk	Chaos Reports (2009)	IT Projects	
		Laudon and Laudon (2000:409)	Information Systems	
		Morisio *et al.* (2007:306)	Software projects	
		Yeo (2002:245)	Information Systems	
	Reliable and realistic estimates	Dalcher and Brodie (2007, 8)	Software projects	
	Good time, cost, and quality factors control	Doi (2006:2)	IT Projects	
		Richard and Lang (2007:3)	IT Projects	
	Project performance	Lind and Culler	IT Projects	

		(2009:6)		
		Nah *et al.* (2001:294)	ERP	
		Schwalbe (2007:17 – 18)	IT Projects	
	Resources	Biehl (2007:55)	Knowledge Based Systems	
		Wixoma and Watso (2001:33)	Management Information Systems	
		Wong (2005:267)	Information Systems	
Clear statement of requirements and specifications	Clear statement of requirements and specifications	Al Neimat (2005:3)	IT Projects	25
		Arntzen and Nkosi (2009:215)	Knowledge Based Systems	
		Biehl (2007:55)	Information Systems	
		Chaos Reports (2009)	IT Projects	
		Dalcher and Brodie (2007: 8)	Software projects	
		Kronbichler *et al.*(2009:15)	ERP	
		Lind & Culler (2009:6)	IT Projects	
		Lu *et al.* (2008:359)	Information Systems	
		Moohebat *et al.* (2010:105)	ERP	
		Nah *et al*. (2001:291)	ERP	
		Nasir and Sahibuddin (2011:2182)	Software projects	
		Ngai *et al.* (2008:1)	ERP	
		Olson and Zhao (2007:136)	ERP	
		Poon and Wagner (2001:396)	Information systems	
		Richard and Lang (2007:2)	IT Projects	
		Yardley (2002:55)	IT Projects	
		Al-Mudimigh *et al.*(2010:5)	Portal implementation	
		Benediktsson (2002:7)	Software Projects	
		Morisio *et al.*	Software projects	

		(2007:306)		
		Poon and Wagner (2001:396)	Information systems	
		Richard and Lang (2007:3)	IT Projects	
		The Standish Group (1995:5)	IT Projects	
		Yeo (2002:245)	Information Systems	
	Strategy and purpose	Wong (2005:267)	Knowledge-based Systems	
	Requirement Analysis	Remus (2006:6)	Portal implementation	
Adequate technology: standard software infrastructure, technical implementation	Standard software infrastructure, technical implementation	Arntzen and Nkosi (2009:215)	Knowledge Based Systems	22
		Chaos Reports (2009)	IT Projects	
		Dalcher and Brodie (2007:8)	Software projects	
		Ehie and Madsen (2005:553)	ERP	
		Futel *et al.* (2002:512)	Software Project	
		Johnson (2000:71)	IT Projects	
		Kronbichler *et al.* (2009:15)	ERP	
		Lind & Culler (2009:6)	IT Projects	
		Nasir and Sahibuddin (2011:2182)	Software projects	
		Remus (2006:6)	Portal implementation	
		Wourms (2004:2)	IT Projects	
		Yardley (2002:115)	IT Projects	
		Yingjie (2005:44)	ERP	
	Technology system works as expected and solves the problems	Doi (2006:2)	IT Projects	
	Technology infrastructure Management	Benediktsson (2002:7)	IT Projects	
		The Standish Group (1995:5)	IT Projects	
	Software project	Benediktsson	Software Project	

	objectives	(2002:7)		
		Humphrey (2005:27)	Software Projects	
	Development technology	Poon and Wagner (2001:396)	Management Information Systems	
		Wixoma and Watso (2001:33)	Information systems	
	Organisational infrastructure	Arntzen and Nkosi (2009:215)	Knowledge-based Systems	
		Wong (2005:266)	Knowledge-based Systems	
User involvement and training	User involvement and training	Chaos Reports (2009)	IT Projects	21
		Chow and Cao (2008:970)	Software projects	
		Dalcher and Brodie (2007:8)	Software projects	
		Kronbichler *et al.* (2009:15)	ERP	
		Laudon and Laudon (2000:406)	Information Systems	
		Lind & Culler (2009:6)	IT Projects	
		Lu *et al.* (2008:359)	Information Systems	
		Morisio *et al.*(2007:306)	Software projects	
		Nasir and Sahibuddin (2011:2182)	Software projects	
		Olson and Zhao (2007:136)	ERP	
		Rainer and Hall (2002:81)	Software projects	
	Incorporate knowledge for users	Sun *et al.* (2005:193)	ERP	
		Tesch *et al.* (2009:663)	Information systems	
		Wixoma and Watso (2001:33)	Management Information Systems	
		Yardley (2002:114)	IT Projects	
		Yingjie (2005:44)	ERP	
		Young and Jordan (2008:719)	Project Management	

	Client acceptance	Al-Mudimigh *et al.* (2010:5)	ERP	
		Belout and Gauvreau (2004:5)	Project Management	
		Olson and Zhao (2007:136)	Portal implementation	
		Remus (2006:6)	Portal implementation	
(Interdepartmental) cooperation and communication	Effective communication	Al Neimat (2005:3)	IT Projects	17
		Arntzen and Nkosi (2009:215)	Knowledge Based Systems	
		Belout and Gauvreau (2004:5)	Project Management	
		Biehl (2007:55)	Information Systems	
		Kronbichler *et al.* (2009:15)	ERP	
		Lind & Culler (2009:6)	IT Projects	
		Lu *et al.* (2008:359)	Information Systems	
		Moohebat *et al.* (2010:105)	ERP	
		Nah *et al.* (2001:291)	ERP	
		Nasir and Sahibuddin (2011:2182)	Software projects	
		Ngai *et al.* (2008:1)	ERP	
		Olson and Zhao (2007:136)	ERP	
		Remus (2006:6)	Portal implementation	
		Tesch *et al.* (2009:663)	Information systems	
		Wourms (2004:2)	IT Projects	
		Yardley (2002:55)	IT Projects	
		Yeo (2002:245)	Information Systems	
Proper project plan	Proper project plan	Al Neimat (2005:3)	IT Projects	14
		Belout and Gauvreau (2004:5)	Project Management	
		Blindenbach-Driessen and van den Ende (2006:557)	Project-based firms	
		Dalcher and Brodie	Software projects	

		(2007: 8)		
		Humphrey (2005:27)	Software Project	
		Lu *et al.* (2008:359)	Information Systems	
		Morisio *et al.* (2007:306)	Software projects	
		Nasir and Sahibuddin (2011:2182)	Software projects	
		The Standish Group (1995:5)	IT Projects	
		Yardley (2002:55)	IT Projects	
		Young and Jordan (2008:719)	Project Management	
	Contingency Plans	Lind & Culler (2009:6)	IT Projects	
	A thorough planning process of requirements	Richard and Lang (2007:2)	IT Projects	
		Yahya at (2002:29)	Information Systems	
Business process re-engineering	Business process re-engineering	Biehl (2007:55)	Information Systems	12
		Ehie and Madsen (2005:553)	ERP	
		Jarrar *et al.* (2010:7)	ERP	
		Kronbichler *et al.* (2009:15)	ERP	
		Moohebat *et al.* (2010:105)	ERP	
		Nah *et al.* (2001:293)	ERP	
		Nasir and Sahibuddin (2011:2182)	Software projects	
		Ngai *et al.* (2008:1)	ERP	
		Remus (2006:6)	Portal implementation	
		Sun *et al.*(2005:193)	ERP	
		Yahya et al. (2002:28)	Information Systems	
		Yingjie (2005:44)	ERP	
Change management	Change management	Biehl (2007:55)	Information Systems	12
		Kronbichler *et al.* (2009:15)	ERP	

		Moohebat *et al.* (2010:105)	ERP	
		Nah *et al.* (2001:293)	ERP	
		Nasir and Sahibuddin (2011:2182)	Software projects	
		Ngai *et al.* (2008:1)	ERP	
		Remus (2006:6)	Portal implementation	
		Tan *et al.* (2009:7)	Information Systems	
		Yardley (2002:55)	IT Projects	
	Changing Requirements & Specifications	Benediktsson (2002:7)	IT Projects	
		The Standish Group (1995:5)	Information Systems	
		Yeo (2002:245)	Software Projects	
Project Champion or ownership	Project Champion or ownership	Dalcher and Brodie (2007:8)	IT Projects	12
		Esteves and Pastor (2002:1077)	ERP	
		Esteves *et al.* (2004:9)	ERP	
		Kronbichler *et al.* (2009:15)	ERP	
		Moohebat *et al.* (2010:105)	ERP	
		Nah *et al.* (2001:292)	ERP	
		Ngai *et al.* (2008:1)	ERP	
		Olson and Zhao (2007:136)	ERP	
		Tan *et al.* (2009:7)	IT Projects	
		Wixoma and Watso (2001:33)	Management Information Systems	
		Yardley (2002:55)	Information Systems	
	Proactive in dealing with problems	Yeo (2002:245)	Information Systems	
Vendor Support	Vendor Support	Doi (2006:2)	IT Projects	10
		Ehie and Madsen (2005:553)	ERP	
		Kronbichler *et al.* (2009:15)	ERP	

		Moohebat *et al.* (2010:105)	ERP	
		Nasir and Sahibuddin (2011:2182)	Software projects	
		Ngai *et al.* (2008:1)	ERP	
		Olson and Zhao (2007:136)	ERP	
		Tan *et al.* (2009:7)	Information Systems	
		Yeo (2002:245)	Information Systems	
	Delivery strategy	Chow and Cao (2008:970)	Software projects	
Systems development methodology	Formal methodology	Avison (2008:35)	Information Systems	10
		Fowler (2001:12)	IT Projects	
		Nasir and Sahibuddin (2011:2182)	Software projects	
		Wourms (2004:2)	IT Projects	
		Yahya *et al*. (2002:31)	Information Systems	
		Yardley (2002:203)	IT Projects	
	Processes and activities	Gai and Xu (2009:563)	Knowledge Based Systems	
		Rainer and Hall (2002:79)	Software projects	
		Wong (2005:266)	Software Project	
	Prototyping	Poon and Wagner (2001:396)	Information systems	
Organizational culture	Organizational culture	Arntzen and Nkosi (2009:215)	Knowledge Based Systems	8
		Gai and Xu (2009:563)	Software Project	
		Moohebat *et al.* (2010:105)	ERP	
		Olson and Zhao (2007:136)	ERP	
		Remus (2006:6)	Portal implementation	
		Wong (2005:266)	Knowledge Based Systems	
		Yardley (2002:129)	IT Projects	
		Yardley (2002:55)	IT Projects	
Software	Software development,	Belout and	Project	7

development, testing, and troubleshooting.	testing, and troubleshooting.	Gauvreau (2004:5)	Management	
		Benediktsson (2002:7)	Software Projects	
		Kronbichler *et al*. (2009:15)	ERP	
		Moohebat *et al.* (2010:105)	ERP	
		Nah *et al.* (2001:294)	ERP	
		Ngai *et al.* (2008:1)	ERP	
		Yardley (2002:115)	IT Projects	
Legacy systems knowledge (data analysis & conversion)	Legacy systems knowledge (data analysis & conversion)	Kronbichler *et al.* (2009:15)	ERP	6
	Data Management	Moohebat *et al.* (2010:105)	ERP	
		Ngai *et al.* (2008:1)	ERP	
		Poon and Wagner (2001:396)	Information systems	
		Sun *et al.* (2005:193)	ERP	
	Data Accuracy	Biehl (2007:55)	Information Systems	
Partnership		Lang (2007:3)	IT Projects	2
		Kronbichler *et al.* (2009:15)		

Table 2.3.2: Critical success factors for IT projects (as per literature study)

2.3.3. The ranking of CSFs as per current study versus the Standish Group Report (2009)

This sub-section is aimed at presenting the CSFs assumed to have a strong impact on the successful development of (IT) projects as per this study. The CSFs were arranged in order of importance, ranged from 1 to 10. We will assess the CSFs and their ratings observed on the Standish Report 2009 against the investigated CSFs in this study.

Table 2.3.3 lists the top ten success factors per frequency of citations from the literature. Furthermore, CSFs are compared to the top ten success factors as per the latest study conducted by The Standish Group Report (2009), and they are arranged in order of importance. The ranking is from 1 to 10. Consequently, these ten CSFs are cited as most important in this literature and they are evaluated from top to bottom. Table 2.3.3 shows the frequency of citations of the evaluated factors; over three years prior to 2011. This is aimed at determining and weighing the value of the underlying factors.

No	Success factors	Frequency of citation (All)	Frequency of citation (2008 - 2011)	Chaos Report (2009) ranking from 1-10
1	Top management support	31	13	2
2	Team composition & teamwork	29	11	8
3	Project management	27	9	7
4	Clear statement of requirements and specifications	25	9	3
5	Adequate technology: standard software infrastructure, technical implementation	22	6	10
6	User involvement and training	21	9	1
7	(Inter-departmental) cooperation and communication	17	8	4
8	Proper project plan	14	4	-
9	Business process re-engineering, Change management, Project champion or ownership and Vendor support	12	5	-
10	Systems development methodology	10	3	

Table 2.3.3: Ranking of critical success factors (as per citation frequency)

Top management support is considered to be the most important success factor in this literature due to its highest frequency citation. Thus it gains the highest and most valuable factor in studies conducted from 2008 to 2011; counted 13 out of 32 citations. It has been evaluated as being the second order of importance as per The Standish Group (2009) findings. Subsequently, team composition and teamwork ranked as the second most important success factor and was evaluated to be the eighth most important factor as per study conducted by The Standish Group (2009). In the third of the previous years of study, it gains 11 counts on citations. User involvement was deemed to be the sixth most important factor (but ranked the first important factor in The Standish Group, 2009). The systems development methodology is the least in the ranking of top ten factors, while it does not count to be critical in the study by The Standish Group (2009). Business process re-engineering, change management, project championing and vendor support hold ninth position as they both came to 12 citations, and as such they are both not regarded as crucial in the Study of Chaos report (2009).

The statistics on project outcomes according to The Standish Group Reports are introduced in the section below to verify the above information.

2.3.4. A review of previous findings on the Standish Group Reports

The Standish Group's purpose since 1994 has been to conduct research on IT projects. Their focus has been to identify (The Standish Report, 1994:2):

- The scope of software project failures;
- The major factors that cause software projects to fail; and
- The key ingredients that can reduce project failures.

Table 2.3.4 shows the figures from the previous studies, and it is evident that there has been a decreasing trend in project success rate as well as a significant increase in the number of failures (The Standish Report, 2001:2; The Standish Report 2009)

Standish Project figures over the years			
Year	**Successful %**	**Challenged%**	**Impaired %**
1994	16	53	31
1996	27	33	40
1998	26	46	28
2000	28	49	23
2004	29	53	18
2006	35	46	19
2009	32	44	24

Table 2.3.4: The Standish Group project results over the years

The figures show that information technology (IT) development project management is in a crisis in view of the fact that the results of Study of Chaos report (2009) shows a marked decrease in project success rates as compared to the previous study Study of Chaos report (2006) with 32% of all successful projects which are delivered on time, on budget, with required features and functions, as compared to the 35% of the Study of Chaos report (2006) results. Furthermore, the study reveals that 44% were challenged (meaning they were late, over budget, and/or with less than the required features and functions) which is less than 46% of Study of Chaos report (2006). Subsequently, 24% failed (which means cancelled prior to completion or delivered and never used) which is a significant increase of failures as compared to 19 % of failures from the previous study. Another point to make is that Study of Chaos report (2009) results signify the second highest failure rate over the decade – 24%

Though many authors have been keen on accepting the subsequent results of The Standish Reports, yet Eveleens *et al.* (2010*:*31) question the veracity of their figures. The basis of their judgement relies on the following aspects:

- **Misleading definition**

The Standish definitions of successful and challenged projects turn out to be a hurdle since their definition of successful projects is deemed to be devoted to an initial forecast of cost, time and functionality (Eveleens *et al.*, 2010*:*31).

- **Unrealistic rates**

The observations of The Standish Group are assumed to be biased due to the fact that there are under-runs for cost and time and over-runs for the amount of functionality (Eveleens *et al.*, 2010*:*31).

- **Perverting accuracy**

The Standish definitions project the cost, time and functionality overestimations which pervert instead of improving the estimation accuracy (Eveleens *et al.*, 2010*:*33).

- **Meaningless figures**

The other major problem is that the Standish figures are meaningless. There is evidence that that large biases occur in practice (Eveleens *et al.*, 2010*:*33).

Sensibly, The Standish Group Reports are meant to examine why IT projects fail and to provide a solution to the problem of increasing project failures experienced in organizations (The Standish Group Reports, 1994*:*8).

2.3.5. The discussion of critical success factors (CSFs)

This section discusses the previously cited critical success factors that, according to this literature, are assumed to be the greatest contribution to information technology (IT) projects' success due to their order of rank (Refer to Table 2.3.2). The judgement is made from the frequency of citations that emerged. The integral thoughts from previous researchers constitute the ranking attributes; range from the most importance down to the less importance. Consequently, the critical success factors are arranged and discussed as order of high ranking as per Table 2.3.2.

2.3.5.1. Top management support

Top management support is defined as a commitment of senior manager, executive or the CEO to evaluate plans and keep track of the project progress and facilitate the management problems likely to arise in the pursuit of the project (Young & Jordan 2008*:*720). A project must gain sponsoring from the executive from the organization so as to keep in touch with the progress of the project and be able to discover the serious difficulties the project may experience during its development (Yardley 2002:115; CHAOS, 2000*:*4, Jarrar *et al.,* 2010*:*7). Top management should justify the new goals, mission and vision; formulate and approve the responsibilities and communicate them to the employees (Nah *et al.,* 2001*:*291).

Most importantly, the intervention of top management brings about effective decisions to resolve conflicts and imbue all the stakeholders with a positive influence in the acceptance of the project (Jarrar *et al.,* 2010*:*7). Noticeably, the project will be vulnerable to malfunction if there is a lack of top management involvement or support for the evolving projects (CHAOS , 2000*:*4).

2.3.5.2. Team composition and teamwork

According to Kronbichler *et al.* (2009*:*16), team composition and teamwork constitute the project team competence, dedicated resources, use of consultants, teamwork and composition, skills and compensation, personnel, appropriate usage of consultants, balanced team, and release of business experts. Certainly training, re-skilling and professional development play an important role in ensuring the success of IT development (Nah *et al.*, 2001*:*293; Al-Mudimigh *et al.*, 2010*:*10).

It is suggested that every stage in development projects requires vigorous training (Al-Mudimigh *et al.*, 2010*:*14; Tesch *et al.*, 2009*:*663). The team member and project leaders have to acquire extensive knowledge about the functionality of the emerged system. Thus, it is vital for the steering committee members to have overview knowledge of the functionalities of the development system. In addition, the training of users of the system plays an important role in getting them to understand the procedures and processes (Al-Mudimigh *et al.*, 2010*:*14).

Consequently, according to Tesch *et al.* (2009*:*663) the study shows that there is a significant relationship between IT project development and both users' knowledge of Information systems (IS) development and developers' knowledge of application domains. Therefore, to a greater extent, training and education make a great contribution to the success of the IT project (Tesch *et al.*, 2009*:*663). In contrast, Al-Mudimigh *et al.* (2010*:*14) argue that it is a challenge to train employees and to devise a strategy for the end-user training. Furthermore, the study conducted by Young and Jordan (2008:720) rejects the notion that competent, hard-working and focused employees are the most critical success factors in IT project success.

2.3.5.3. Project management

Project management integrates the application of skills and knowledge (Kronbichler *et al.,* 2009*:*16). The PMI (2004:8) proclamation is that "project management is the application of knowledge, skill, tools and techniques to project activities to meet project requirements".

Project management practice includes early discovery of issues, opportunities, and potential project pitfalls which will most likely warranty the project success (Richard & Lang 2007:3). Ultimately, the measure of project success will be determined by the manner in which the project manager will control scope, time, cost and quality factors (Schwalbe, 2007:10, Richard & Lang, 2007:3).

The project management principles affirm that project managers proactively possess a high sense of evaluation in terms of the reasonableness of project and schedule timelines; analyse the needs for the project resources; and develop project and risk mitigation strategies (Richard & Lang 2007:3; Schwalbe, 2007:14).

Practically, Yardley (2002:115) stipulates that project management includes the following fundamental factors that are thought to be a positive contribution to the success of the project, viz. impact of project on business, achievement of timescale, quality and cost. In view of that, good project management is important for the success of IT projects (Nah *et al.,* 2001*:*292).

2.3.5.4. Clear statement of requirements and specifications

To obtain a clear statement of requirements and specifications will trigger the clear vision and goal of the projects. It is vital for project managers to devise effective measures on how each project will contribute to the goals and objectives of the organization (Ngai *et al.,* 2008:8, Richard & Lang 2007:2; Kronbichler *et al.,* 2009*:*16). It is recommended that the definition and redefinition of the final goal and vision of the end-product in which the project outcomes will merge with the long-term objectives of the organizations, including a justification for the investment should be declared (Richard & Lang, 2007:2; Ngai *et al.,* 2008:8; Yardley, 2002:116).

To identify the project the goals and mission is a strong reason to correspond deliberately with all the stakeholders, including the end-users. Predominantly, this factor ensures the success of the project (Wourms, 2004*:*2). In addition, this endeavour will anticipate the opportunities, risks, cost and resources attached to the adoption of the emerged system (Richard & Lang, 2007:2; Ngai *et al.,* 2008:8) and thus ensure keeping the project focused on business benefits (Nah *et al.,* 2001*:*291).

Generally, project and organizational goal alignment opportunities will outline the metrics which will determine the success of the project (Yardley, 2002:116).

2.3.5.5. Adequate technology: standard software infrastructure, technical implementation

The application of standard software infrastructure enables the development team to give attention to the actual business principles instead of technology itself (CHAOS, 2000:6)

It is to the benefit of the project for the project manager to have sufficient knowledge of the technology of both the software and hardware intended to be used on a project. This will provide the advantage of knowledge transfer among the team members. The other advantage is that a manager will have a proper technical lead to enable him to make realistic estimates, working within the schedule and providing correct information to impact on the project successfully (Wourms, 2004:2).

Furthermore, the knowledge of managers of technology promotes management visibility in that the manager will see whether the developers are doing what they are supposed to do regarding the technical implementation and thus be able to lend a hand to produce the required results (Doi, 2006:2). As a result, lack of experience with technology to be used in a project is a high risk which could cause the project to be more costly and run behind schedule (Laudon & Laudon, 2000:409).

In terms of organizational infrastructure, project managers must develop a sense of organizational ownership to determine the effect of the outcome of the emerging project along the strategic objectives of the organization (Richard & Lang, 2007:2). In particular, project managers must devise adequate methods of management for each sub-project (Yardley, 2002:103).

This endeavour, when practised in the early process, will reveal the opportunities and potential pitfalls that may arise during the project development (Richard & Lang 2007:3). Consequently, the characteristics of technology within an organization have a great contribution to the development of IT projects (Ngai *et al.*, 2008:8). To successfully undertake the project, adequate hardware and software infrastructure is required to ensure appropriate implementation.

2.3.5.6. User involvement and training

Projects include users and stakeholders who have expectations and needs, and in order to meet their needs, their opinions have to be considered extensively to ensure the success of the project (Belout & Gauvreau, 2004:1; Young & Jordan, 2008:715). Thus, projects should be able to solicit users' involvement appropriately in order to establish and manage the project requirements and expectations appropriately (Young & Jordan, 2008:715).

It is vital that project managers should have strong "early" bonds with key system users and front-line personnel to formulate the goal to which the project should aspire (Anderson & Birkely, 2005:527; Richard & Lang, 2007:2; Yardley, 2002:55). Furthermore, the project manager must smooth over the entire process to meet the needs and the expectations of users or people affected by project activities (Schwalbe, 2007:10; CHAOS, 2000:4). In view of that, heavy user involvement during systems development is more beneficial in patterning the system in a way satisfactory to all people involved or affected by the system, in that it makes it easier for them to control the outcome and moreover, to meet the business requirements (Laudon & Laudon, 2000:406). Most importantly, users become much more ready to accept the project outcomes if they have been participating in the development change process.

2.3.5.7. (Interdepartmental) co-operation and communication

One of the crucial success factors to the success of IT projects is the effective communication and hence interdepartmental co-operation among the stakeholders (Kronbichler *et al.,* 2009:16).

It is of great significance to establish an understanding and approval for systems development (Kronbichler *et al.*, 2009:16). Hence, the most important competency of the project manager is the ability to communicate with stakeholders, either in one-on-one sessions, in small groups, and before large audiences (Wourms, 2004:2; Nah *et al.,* 2001:291). Additionally, to satisfy the users' information needs, it is required that there must be an open and honest information policy that should be communicated among all (Sarker & Lee, 2003:814).

A project manager must possess strong listening skills and be able to show empathy and bestow recognition and praise on project team members (Wourms, 2004:2). This will not only guarantee the earning of respect from the team, but will provide early warning of problems that may crop up unexpectedly to impact on the success or the failure of the project. It is evident from some previous research that effective communication is widely recommended in organizations for the achievement of success of the project since there will be a thorough exchange opinions and information during the development of IT projects (CHAOS, 2000:8).

2.3.5.8. Proper project plan

The study conducted by Young and Jordan (2008:715) proves that with a more formalized high-level plan, the project is likely to accomplish its goals and objectives; however, the same study indicates that there is no sufficient evidence that high-level planning is the most important critical success factor. Successful project planning should be a good document that articulates and identifies the underlying business problems and predicted benefit to the business if the problem is to be resolved (Yardley, 2002:117).

To further unpack the concept, the clearly-defined and refined requirements allow the project to be easily planned and managed. Thus, the more refined the requirements, the less the risk and the higher the chance the project will succeed (Laudon & Laudon, 2000:409).

2.3.5.9. Business process re-engineering (BPR)

In order to improve the functionality of technology to fit the organization's needs, an organization should re-engineer the business processes to be in line with the technology instead of trying to modify the software to fit the organization's current business processes (Ngai *et al.,* 2008:4; Jarrar *et al.,* 2010:9). Business process re-engineering is a crucial factor in IT project success in view of the fact that technology tailoring alone cannot ensure customer satisfaction (Cheng & Chiu, 2008:258). Therefore, re-engineering should be carried out iteratively to adapt the advancements from the emerged technology (Nah *et al.,* 2001:294).

Certainly, an adequate degree of common fit between the organization and the technology to carry out the systems development is critical to the success of the project (Ngai *et al.,* 2008:4). Although the business re-engineering may be perceived to be risky, it imparts flexibility to the organization to accommodate innovative ideas to better the customer service (Ngai *et al.,* 2008:4). The study conducted by Jarrar *et al.* (2010*:*9) proves that modifying the business process to be compatible with software is a crucial endeavour to ensure the success of the project.

2.3.5.10. Change management

This is one of the critical success factors and includes change management, change management programmes and culture and commitment to change (Kronbichler *et al.*, 2009*:*18). Management of change has become an integral issue in the development of IT projects due to the impact of new technologies, systems, business rules and processes (Yardley, 2002*:*36). Training and education of employees is part of the critical part in changing management (Ngai *et al.*, 2008:4). Thus, embarking on an advanced IT system demands procedural changes which will eventually embrace physical changes (Al-Mudimigh *et al.*, 2010*:*10). In addition, a culture which embraces shared values and common aims is prone to success (Nah *et al.*, 2001*:*293).

Given that fact, change management has an effect on the success of the project as it allows the recognition of the opportunities and potential problems that need urgent attention (Al-Mudimigh *et al.*, 2010*:*10). Therefore, it is profitable to evolve change management in the initial phase throughout the entire development of the project (Nah *et al.*, 2001*:*293).

Sensibly, according to Al-Mudimigh *et al.* (2010*:*9), the study shows that companies are reluctant to change due to their resistance to learn new IT products. As a result, it is assumed that resistance to change is most likely to hinder the success of projects (Al-Mudimigh *et al.*, 2010*:*10; Yardley, 2002*:*36).

2.3.5.11. Project champion

The project champion is defined as the person who is committed to supporting the project to its completion, thus taking on ownership and responsibility to acquire the project resources (Esteves & Pastor, 2002*:*1077; Esteves *et al.,* 2004*:*11). A project champion should be a high-level executive sponsor with the authority to 'champion', monitor and control the project across the organization (Ngai *et al.*, 2008*:*9; Esteves & Pastor 2002*:*1083; Esteves *et al.,* 2004*:*9; Young & Jordan, 2008*:*715). The regular existence of executive to pursue resolve stability is critical to success of the IT project (Nah *et al.,* 2001*:*292). Hence the executive sponsor's support to IT projects is deemed to be critical for the project to achieve its goal (CHAOS, 2000*:*9; Esteves & Pastor, 2002*:*1077).

2.3.5.12. Vendor support

Vendor support includes vendor support, vendors' tools and the use of vendors' development tools (Kronbichler *et al.*, 2009*:*16). The vendor is the entity that supplies the project with products (Doi, 2006*:*2). The stakeholders, which include the programme managers, procurement manager, developers, and users, are envisaged to take the responsibility to control and support the vendors (Doi, 2006*:*2).

In particular, the procurement manager who accounts for vendor control and support is obliged to catch up with the rapid changes of technological trends (Doi, 2006*:*2). Certainly, one of the main reasons IT projects fail is the failure to control the vendors appropriately as they are not given enough attention to enable them to develop the success system (Doi, 2006*:*3; Ngai *et al.*, 2008*:*8; Kronbichler *et al.*, 2009*:*16). As a result, the organization needs to do thorough research about vendors in order to improve decision making support with respect to IT projects development (Lee & Lee, 2001*:*207).

2.3.5.13. Systems development methodology

Information systems methodologies are categorized into two main areas: project management and systems development. The purpose of project management methodologies is to institute control over resources, tasks, budgets, and risks within the project; and they provide a strategy to underpin the development of IT projects (Yardley, 2002:80).

Typically, methodologies embrace a disciplined process that underpins the planning of the development of software, and this endeavour will ensure predictable, resourceful and proficient development (Fowler, 2001:1). As a result, according to Chaos (2001:5), formal methodologies have a great impact on the success of IT Projects.

However, Fowler (2001:1); Young and Jordan (2008:720) believe that methodologies have not been recognized as a popular critical success factor to the IT projects success. As such, they have been criticized as being bureaucratic in that there are too many steps to follow. Hence, the study conducted by Young and Jordan (2008:720) does not support methodologies as the critical factor to IT projects.

With respect to evolutionary development methodologies, an evolutionary development methodology is regarded as a key factor for system success. This factor is achieved through the utilization of prototyping which articulate that new technology can present value to the development of the projects (Poon & Wagner, 2001:396).

In general, the processes and methods define the practices that can be performed with knowledge of the organization's IT projects. The endeavour to execute processes has a positive influence on the systems development success. Thus, it is important that organisations adopt a process-based view of the system (Wong, 2006:271).

2.3.5.14. Organizational culture

Organizational culture defines the core beliefs, values, norms and social customs to guide the individual's acts and behaviours in an organization. As such, an organizational culture emphasises a high sense of mutual trust (Gai & Xu, 2009:563). Certainly, organizational culture involves the degree of involvement, transparency, shared vision and goals which form an integral part in the success of systems development and is therefore considered to be critical to the success of IT project development (Bechina & Ndlela, 2008:215).

Nonetheless, the organizational culture of the organization such as views, beliefs, norms, and unwritten rules is likely to hinder the requirements of the management process that allows the elicitation of requirements (Yardley, 2002:72).

2.3.5.15. Software development, testing and troubleshooting

This critical success factor comprises software development, testing and troubleshooting along with the main aim of eliminating troubleshooting (Kronbichler *et al.*, 2009:17). Software development, testing and troubleshooting all need to be practised from the initial phase of the project to ensure the proper configuration at each stage of the project (Nah *et al.*, 2001:293). In essence, testing and troubleshooting are aimed at verifying that software functions performed as initially purposed (Ngai *et al.*, 2008:9).

According to Nah *et al.* (2001:294); Ngai *et al.* (2008:4), software development, testing and troubleshooting are critical factors for the success of IT projects. Nonetheless, particularly in an ERP system, its co-operation with legacy systems should be handled thoroughly since it is a hurdle in carrying out the integration (Ngai *et al.*, 2008:9).

2.3.5.16. Legacy Systems Knowledge

Legacy Systems Knowledge involves critical success factors namely business and IT legacy systems, adequate legacy systems knowledge and data analysis and conversion (Kronbichler *et al.*, 2009:18). To access reliable and consistent data is a key factor to the success of the systems development (Poon & Wagner, 396),

As a consequence, to manage the complexity of development projects, business and legacy systems are an important part of a successful IT project (Nah *et al.*, 2001:289). However, according to Ngai *et al.* (2008:4), legacy systems knowledge counts as the least critical factor in the development of IT projects.

2.3.5.17. Partnership

Partnership embraces trust between partners, consultants, vendor, and customer partnership (Kronbichler *et al.*, 2009*:*18). According to Richard and Lang (2007:3), it is essential for project managers to cultivate firm relationships among the IT project team, users, and stakeholders to maintain and re-evaluate the needs and expectations. This relationship will have a positive effect on the development of IT projects.

In essence, securing these relationships in the initial development of the process would facilitate the process that serves to identify the potential areas of conflict and risk (Richard & Lang, 2007:3). As result, partnerships need a high sense of contact to encourage regular meetings among the business stakeholders (Nah *et al.*, 2001*:*289) which will of necessity have a great influence to the success of IT projects.

2.4. Conclusion

The chapter contained a literature review of the relationship between systems development methodologies (SDMs) and the success of information technology (IT). Furthermore, it explored the different critical success factor (CSFs) and their impact to the IT projects success. The summary for both contexts is presented below.

2.4.1 Section One: Systems development methodologies (SDMs)

The purpose of this section was to bring about an understanding of the role of systems development methodologies in IT project success. Particularly, the focus has been on ascertaining the usage of systems development methodologies in information systems development in organisations, its advantages and benefits, the disadvantages and critics, and finally, to identify and discuss different types of systems development methodologies which, according to previous empirical researches envisaged are more popular in the organisations.

Various definitions of systems development methodology have been suggested. However, the term 'methodology' is not well-defined either from the reviewed literature or by practitioners (Avison & Fitzgerald, 2008:34). Most often it is used interchangeably with terms such as method, software development methodology, systems development methodology, process models, technique and tools (Yahya *et al.,* 2002:15; Strode *et al.,* 2009:1). To make sense of the role of SDMs in IT projects, the study adopts and articulates the view of Huisman and Iivari (2006*:*32) that SMDs constitute four elementary concepts which are: the philosophical approach, methods, process models and tools and techniques.

In sequence, ten types of SDMs namely*:* SSADM, IEM, RUP, XP, ETHICS, SSM, YSM, JSP, and WISDM were outlined and discussed according to their components namely philosophical approach, method, process model, and tools and techniques as per the view of Huisman and Iivari (2006*:*32) of the importance to explore SDMs. Substantially, the types of methodologies vary from one project to another. This means that organizations may use more than one methodology for several projects (Yahya *et al.,* 2002:26).

The knowledge gained from the study is that one among other underlying success factors is assumed to be the systems development methodology (SDM). Hence, the use of SDM is always regarded as an improvement of the quality and productivity in systems development (Yahya *et al.,* 2002:15; Mihailescu & Mihailescu, 2009:1; Griffin & Brandyberry, 2008:2; Huisman & Iivari, 2006:33).

Furthermore, the aforementioned potential benefits of systems development methodologies were depicted. The contention of Kiely and Fitzgerald (2005, 1) and Yaghini *et al.* (2009:3), is that systems development methodologies cannot be approved without being tailored and engineered to fit in with individual development projects. Practically, methodology does not at all times satisfy the needs of the organization. Thus they adapt the methodology to fit their needs (Fitzgerald *et al.,* 2002*:*194).

Another point to make is that developers and IS managers have different viewpoints and expectations about systems development methodologies in that IS managers perceive the support of systems development methodology and their impact to be very optimistic (unlike the developers). As a result, perceptions in congruence among IS managers and IS developers absorb their expectations, assumptions, and norms diversely (Huisman & Iivari, 2006:41).

In conclusion, although information systems development methodology (ISDM) has always been considered to be an integral part of the majority of development projects, and are useful and helpful to organisations, practitioners are still reluctant to adopt them. Hence, Barry and Lang's (2003:220) opinion is that the reason for a low adoption rate of formalized methods is their perceived technical limitations rather than any philosophical objections. Furthermore, it is worth noting that formalized methods were too costly, as a result, many studies about ISDM have been conducted to uproot its usefulness as well as other aspects including its failure.

2.4.1. Section Two: Critical success factors (CSFs)

The purpose of the section has been to present the different success factors which are assumed to be the great contributors to the success of IT projects. The chapter assembles the evaluation of the ranking of critical success factors from which the impact of the success of IT project is measured. Sensibly, the entire evaluation is drawn from the various authors' perspectives.

Furthermore, the reflection on perceptions of different factors according to different authors triggers the construction of the most-stated critical factors as characterized in the literature, and then ended up by listing the factors that as per frequency of citation were regarded as important and related to the success of information technology. The list rates from top to down in order to symbolize the order of importance for the underlying critical success factors. The manifestation perception about the selected factors is measured against the studies (from 2008 to 2011) except the citation of The Standish Report 1994 to determine the weighting value.

Moreover, the chapter explores the subsequent annual findings of The Standish Reports. The study investigated and presented the factors that mostly influence IT project success, challenges and impairments. Typically, the phenomenon of these practices enables the IT project failure or challenge (the Standish Group 2001:2). The Standish Report (2009) proves that there is a great incidence of project failure rates and the study disclosed that only 32% of all projects succeed (meaning that they are delivered on time, on budget, with required features and functions). This confirms a downturn in success rates compared to the previous study, which records 35% success rates and also shows an increase in project failure rates.

This chapter also underlines the top ten factors observed to be the most critical success factors in this literature from which they have been determined according to the number of citations. In the final instance, the chapter lists seventeen critical success factors drawn from literature so as to demonstrate their impact on the success of IT projects.

In my opinion, while similar studies about the success of the IT projects have been conducted, much still has to be done to solicit the IS personnel and IS users' views about success in improving the best practices in order to support the fundamental goals of emerging IT projects.

In particular, the main focal point of this study is to investigate the role of systems development methodologies in Information Technology (IT) projects success. Though not popular, it is evident from literature that a systems development methodology is one of the critical success factors needed for success in information technology (IT) projects. As a result, we hope to overcome previous limitations of similar studies by bringing about knowledge of different systems development methodologies which affect the development of IT projects.

In conclusion, based on a review of IT project success factors in the literature, the study outlined the following 17 success factors as the key practices that influence IT project success. They are listed in order of importance as per the current literature study.

No	Critical success factor	No of citations in this literature
1	Top management involvement and support	32
2	Team composition and teamwork	29
3	Project management	27
4	Clear statement of requirements and specifications	25
5	Adequate technology*:* standard software infrastructure, technical implementation	22
6	User involvement and training	21
7	(Interdepartmental) cooperation and communication	17
8	Proper project plan	14
9	Business process reengineering	12
10	Change management	12
11	Project championing or ownership	12
12	Vendor support	12
13	Systems development methodology	10
14	Organizational culture	8
15	Software development, testing, and troubleshooting	7
16	Legacy systems knowledge (data analysis & conversion)	6
17	Partnership	2

Table 2.4: List of CSFs presented in order of importance as per this study

Chapter 3 - Research Design

3.1 Introduction

The aim of this chapter is to discuss and illustrate the research methodologies employed in this study. A research project needs a design or a structure from which the data collection, data analysis and data interpretation can emanate to construct knowledge in such a context (Mertens, 2005:2; Darren & Lindsey, 2007:60). As such; the research design ensures that the outcome of the study has addressed the fundamental research questions which drive the conclusions of the study to make sense of the knowledge in an explicit and accurate manner (Darren & Lindsey, 2007:60).

To give an overview of the research design, the study articulates the description of the research process and the way which they were exercised in this study. Henceforth, the succeeding sections discuss the different paradigms and eventually underpin the relevant paradigm with its underlying strategies upon which the current study was based in seeking to answer the basic question, viz. that "Is there a relationship between system development methodologies and the success of IT projects?" In this study I would first like to provide an understanding of what a paradigm is and the effects of its fundamental assumptions.

The chapter is categorized as follows:

- Research paradigms.
- Research strategies.
- Data generation methods.
- Data analysis approaches.
- Summary
- Conclusion.

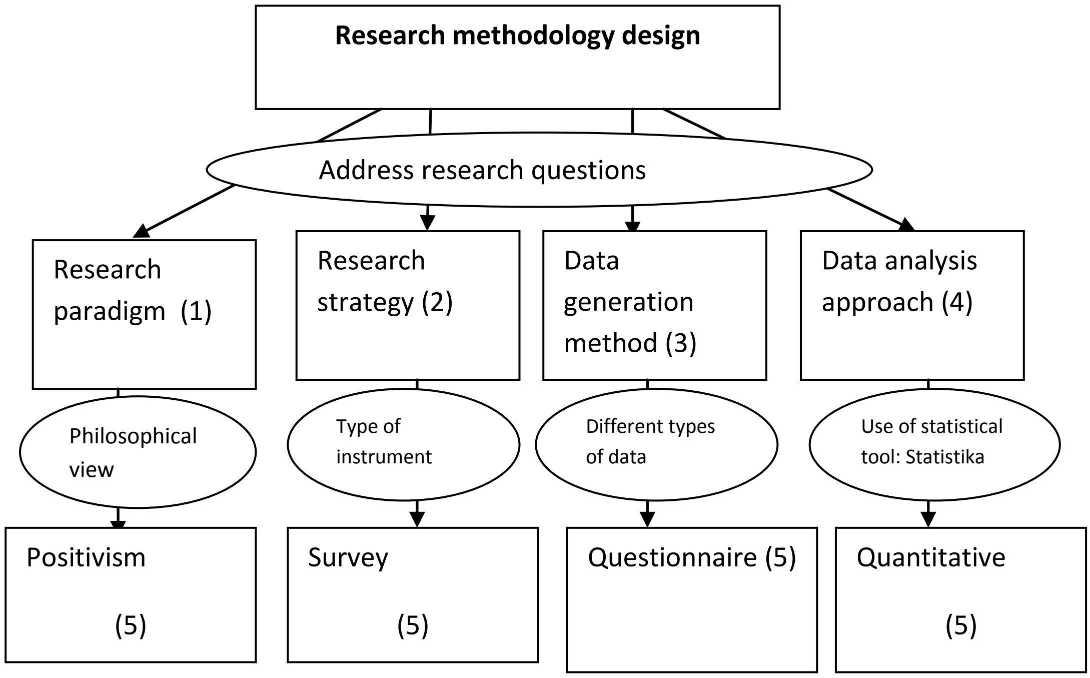


Figure 3.1: An outline of research design

3.2 Research paradigms

According to Johnson and Onwuegbuzie (2004:24) and Oates (2008:282), a research paradigm is a set of common beliefs, values and assumptions of society of researchers concerning the way in which the research is conducted and the way it goes along with its natural world. Similarly, a paradigm is the way people perceive aspects of the world. A philosophical assumption of the nature of reality is crucial in making known the perspective upon which the study is constructed and performed (Krauss, 2005:755). All the philosophical paradigms are grounded on the following assumptions (Johnson & Onwuegbuzie, 2004:24):

Ontological beliefs: Ontology specifies the different views engendered by different philosophical paradigms with regard to the nature and the realities of the world (Oates, 2008:282; Krauss, 2005:754; Mertens, 2005:4). According to Krauss (2005:760) qualitative researchers are generally employed in relativistic and constructivist anthologies which declare that there is no objective reality.

Epistemological beliefs: Epistemology affirms that there are different ways in which knowledge about the nature of the world can be acquired (Oates, 2008:282; Krauss, 2005:754). An epistemological assumption underpins the relationship between the researcher and the participants in the study (Mertens, 2005:4).

In summary, we have learned that a paradigm is the set of philosophical assumptions underpinning research. It is believed that the concept *paradigm* should be regarded as the first step in informing the subsequent selection regarding the methodology, methods, literature or research design. Consequently, a paradigm influences the way in which knowledge is studied and interpreted, and from which point of view; it is the choice of paradigm that sets down the intent, motivation and expectations for the research.

The subsequent section is aimed at presenting a discussion of different philosophical paradigms of significance to the research project. They include the positivist paradigm, the interpretive paradigm and the critical paradigm. The study adopts the positivist paradigm as a result of these considerations; the section will go into extensive detail on its application.

3.2.1 A discussion of different paradigms

3.2.1.1 The positivist paradigm

The positivist paradigm predominates in scientific research (Krauss, 2005:760; Oates, 2008:282; McKelvey, 2002:3; Matveev, 2002; Darren & Lindsey, 2007:64) carried out in a natural sciences domain, e.g. chemistry, computer sciences etc. (Oates, 2008:282). It is less suited to studying the social world, such as people, the way they think and act in the world. A scientific research project seeks to find the regularly occurring patterns or the laws in the realities of the world (Krauss, 2005:760; Oates, 2008:282). Furthermore, the positivist paradigm's concern is that there is an objective truth in the world that can be measured and explained scientifically (Matveev, 2002; Darren & Lindsey, 2007:60).

a) Basic techniques of scientific method in the Positivist paradigm

- **Reduction**: This approach makes provision for the breaking down of complex things into small components to allow ease of study. Gradually hypotheses that have been examined can be compounded into one overall theory (Oates, 2008:284; Darren & Lindsey, 2007:64).
- **Repeatability:** This technique encourages the researcher to repeat the survey or experiment to ensure that the first results were not a fluke or coincident. The experiment can be continual to ensure that the same results will be obtained (Oates, 2008:284).
- **Refutation:** It underlines the notion that if the results are repeated and similar results are not achieved then they disprove the hypothesis. If a hypothesis does not hold true for many scenarios then it can be refute (Oates, 2008:284).

b) Characteristics of Positivism

- **The world exists independently of humans:** There is a physical and social world that exists and is ready to be studied, captured and measured. This is not dependent on human minds (Oates 2008:286; Kim 2003:11).
- **Measurement and modelling:** Insight in the world is discovered by making observations and measurements and producing models (hypothesis, theories) of its performance (Krauss, 2005:760; Matveev, 2002; Oates, 2008:286; Mertens, 2005:7).
- **Objectivity:** The researcher is neutral and objective, not biased. Knowledge is discovered independently of personal values or beliefs (Oates, 2008:286; Matveev, 2002; Kim, 2003:11).
- **Hypothesis testing:** Research is based on empirical testing of theories and hypothesis likely to result in confirmation or disproving of them through the use of statistical analyses (Kim, 2003:11). Scientific hypotheses are value-free; the researcher's own values, biases, and subjective preferences do not have effect in the quantitative approach (Matveev, 2002; Oates, 2008:286).
- **Quantitative data analysis:** In quantitative research, the researcher’s aim is to determine the relationship between an independent variable and a dependent variable in a population. It is most often used in the context of the positivist paradigm (Matveev, 2002; Oates, 2008:286).

- **Universal laws:** Research looks for generalizations - regularly recurring patterns and approved facts regardless of the researcher or the occasion (Krauss, 2005:760; Matveev, 2002; Kim, 2003:11).

c) Judging the quality of positivist research

- **Objectivity**: The research must be free of researcher bias and distortions, and as a result, the findings must be independent of the researcher's personal views and beliefs (Matveev, 2002; Oates, 2008:287; Kim, 2003:11).
- **Reliability**: The research instruments must be neutral, accurate and reliable. The instrument may be equipment used in an experiment, a set of questions, or a researcher conducting interviews (Matveev, 2002; Oates, 2008:287; Kim, 2003:12).
- **Internal validity**: The research should be well-designed so that the researcher examines the right data; similarly, the relevant data should be collected from the right source and be based on how things relate to the real world (Sale *et al*., 2002:45; Matveev, 2002; Oates, 2008:287; Kim, 2003:12).
- **External validity**: Research findings should be generalizable to different people, settings or times. It is aimed at searching for general laws, patterns not finding something unique to one particular case (Matveev, 2002; Oates, 2008:287; Kim, 2003:12).

d) Criticism of the positivist paradigm

- **Reductionism:** It is observed that sometimes reduction might be implausibly practical to some research projects or it may miss an overall picture (Oates, 2008:288). As such, theories will be irrelevant if they can all be defined by reduction with respect to the measurement aspect (McKelvey, 2002:4).
- **Repetition:** It brings forth the notion that it is not always viable to repeat research several times and with different researchers (Oates, 2008:288).
- **Generalization is not always desirable:** Concentrating only on regular laws and generalizable patterns makes one miss out on many things. It is sometimes helpful to study the particular and the unique (Oates, 2008:288).
- **Not everyone sees the world in the same way:** The researcher needs some ways to study how individuals perceive their world and interpret it (Oates, 2008:288). Regular laws and patterns may appear to be observable

in a social world, however, they are assumed to serve as a construction necessary to the people. Regular laws are formed by shared views of people but they are not laws of nature. The researcher also needs some ways to study how groups, organizations, cultures and societies perceive their world and interpret it (Oates, 2008:288; Kim, 2003:10).

We have learned that the main objective of the positivist approach is to test a theory or describe an experience through observation and measurement in order to predict and control forces that surround the world. Additionally, a positivist approach associates with the scientific research (Krauss, 2005:760) and is based on the rationalistic (Mertens, 2005:7).

The next section provides a discussion of the interpretive paradigm, its meaning and effect on the research environment.

3.2.1.2. The interpretive paradigm

The interpretive research paradigm seeks to give an overall understanding of social context, beliefs, personal values, principles, preference and historical background (Darren & Lindsey, 2007:64; Kim, 2006:10; Mertens, 2005:15). Thus, it projects the way in which people develop and construct the social process, and the way which it influences, and is influenced by (Oates, 2006:292). The approach of the Interpretive paradigm towards research is that reality is socially constructed (Mertens, 2005:12; Oates, 2008:292). Hence, it focuses on and emphasizes a subjectivist approach to studying social phenomena and the complexity of human sense-making as the situation develops (Kim, 2003:10). It also ties in with the notion that social reality is viewed and interpreted by individuals according to the ideologies they may hold (Oates, 2008:292).

The Interpretivist paradigm is most likely based on qualitative data collection methods and analysis; alternatively, one could consider the combination of both qualitative and quantitative methods, an approach that is termed a mixed method (Mackenzie & Knipe, 2006).

3.2.1.3. Critical research

The purpose of critical research is to identify power relations, conflicts, and contradictions, and authorize people to serve as the main sources of alienation and domination to harmonize state of affairs (Oates, 2006:296; Mertens 2005:15). It seeks to discover social injustice through which people can change to convey harmony (Kim, 2003:10).

The critical research assumption is that a positivist approach to research fails to address issues of social justice and marginalized peoples (Creswell, 2003:9). In a similar way, critical researchers criticize interpretivist researchers in terms of their failing to foresee some pattern of authority and power that regulates the perception about the world in a particular manner (Oates, 2006:296).

The critical approach, as in the case of positivism, applies qualitative and quantitative data collection and analysis methods (Mackenzie & Knipe, 2006; Mertens, 2005:3).

From this viewpoint, it is assumed that critical research is practised less often than positivist and Interpretivist research; hence, there have been fewer suggestions that criteria should be evaluated. As such, it is assumed that interpretive and critical methods are not as stable as positivist research (Sale *et al.,* 2002:45).

In summary, the table below (Table 3.2) specifies the characteristics of the positivist paradigm, the interpretive paradigm and the critical paradigm. The framework has been developed using language in a range of research texts and grouped so as to match different paradigms discussed above (Mackenzie & Knipe, 2006).

Positivist/ Post-positivist	Interpretivist/ Constructivist	Transformative/Critical
Experimental Quasi-experimental Correlation Reductionism Theory verification Causal comparative Determination Normative	Naturalistic Phenomenological Hermeneutic Interpretivist Ethnographic Multiple participant meanings Social and historical construction Theory generation Symbolic interaction	Critical theory Neo-Marxist Feminist Critical Race Theory Ferrean Participatory Emancipatory Advocacy Grand Narrative Empowerment issue-oriented Change-oriented Interventionist Queer theory Race-specific Political

Table 3.2: Different paradigms adopted from Mertens (2005:2) and Creswell (2003)

Consequently, the next section deals with the research paradigm approach used in this study and furthermore discuss the paradigm's contribution towards answering the fundamental research questions and lastly, provide the reason for using the selected paradigmatic method.

3.2.2. The research paradigm used in the current study

The objective of this section is to present the research paradigm approach employed in this study. The study adopts the positivist paradigm to develop an emergent view of the realities to be dealt with. A positivism paradigm is associated with scientific research, and it is aimed at accomplishing observations and measurements (Oates, 2008:263; Krauss, 2005:760; Matveev, 2002; Mertens, 2005:7). The primary goal in this study was to seek to understand the relationship between a systems development methodology and IT project success and to learn about other critical success factors that influence IT project success. The study attempts to answer the following research questions:

- Does a relationship exist between systems development methodologies (SDMs) and information technology (IT) project success?

- To what extent do systems development methodologies (SDMs) influence information technology (IT) project success? Which other critical success factors influence the success of information technology projects? To what extent do CSFs influence the success of information technology projects (determine their weighting/ranking)?
- What effects do systems development methodologies have as compared to other factors contributing to information technology (IT) project success?
- What perspectives do system developers and team managers develop during their experience of the application of systems development methodologies (SDMs) in the development of an IT system?

The positivist approach is best suited to answer these questions since it determines universal laws, regularities and patterns using valid questionnaires to verify the evidence (Krauss, 2005:760; Oates, 2008:282). Practically, to examine the relationships between dependent variable (SDMs) and an independent variable (IT project success) the scientific methods assumed to be relevant in order to conduct the empirical tests to offer the valid evidence to approve or refute the hypotheses. This is made possible by using quantitative analysis. Thus, positivism underpins the quantitative analysis (Krauss, 2005:760; Oates, 2008:263; Matveev, 2002; Sale *et al.,* 2002:45).

The study will satisfy the following positivist criteria for accuracy:

- Objectivity: we did not have any influence on a participant's view as the facts acquired were independent of the participant's personal perceptions.
- Internal validity: the research design was premeditated and chosen precisely so as to capture the relevant data from appropriate sources.
- External validity: the questionnaires were sent to different people having different roles in different business sectors. The reason was to determine trends and patterns from more than one scenario.
- Reliability: the research instrument was the questionnaire comprising a set of questions which are neutral, accurate and reliable. Most of the questions were adapted from previous successful studies such as Huisman (2000).

3.2.3. The reason for choosing a positivist research approach

- Seeking to investigate the role of system development methodology in the success of an IT projects, a regularly recurring pattern to justify the relationship is required (Krauss, 2005:760; Oates, 2008:282).
- A positivist paradigm is associated with quantitative analysis which uses statistical analysis and mathematical modelling (Oates, 2008:38; Krauss, 2005:760; Matveev, 2002).
- To allow descriptive analysis particularly means to determine the most effective critical success factors and the most favourable system development methodology.
- Consequently it makes possible a logical and objective means to analyse observations and results (Oates, 2008:286).
- Positivist research assumes that the data does not change and its analysis is value-free, as such, it recognizes that a researcher is neutral and objective in terms of the outcome (Oates, 2006:286; Matveev, 2002; Kim, 2003:11).
- The scientific method and its underlying positivist philosophy have been very successful up to now (Oates, 2006:288), hence, positivist research is deemed to be a dominant paradigm (Krauss, 2005:760).
- The study explores a ‘realistic’ ontology instead of giving an impressionist perspective by focusing on participants’ utterances or behaviour, in construct, the focal point is the perception they provided through the questionnaire instrument.

Sensibly, interpretive and critical paradigms are thought to be not as stable and predominant in the scientific domain research as positivist research (Sale *et al*., 2002:45); hence there was a strong conviction that the positivist paradigm is an adequate and applicable approach to provide the scientific measurement and observation (Krauss, 2005:760) to refute or approve the relationship between system development methodology and IT project success.

The subsequent section aspires to assess the description of different strategies that impact on research projects. These strategies are the overall approach to answer the research question (Oates, 2008:35). Eventually, the section will provide a comparison of different strategies.

3.3. Research strategies

3.3.1. The research strategy employed in this study

The study employed the survey research strategy. This section deals in some depth with the survey research strategy dimensions. The *survey* is one of the research strategies associated with the positivist paradigm (Oates, 2008:35). As we have learned, this study employs a positivist paradigm, and therefore knowledge about the survey research strategy becomes relevant and important. However, the positivist paradigm includes the following other strategies, which are design and creation, experiments and case study (Oates, 2008:35). To bring about an understanding of the survey research strategy, its definition, planning and designing and evaluating survey-based research will be provided.

3.3.1.1 Definition

The notion behind the survey is that the same kinds of data from a large group of people will be obtained in a standardized and systematic manner through which the main concern is to draw the pattern and trends from a large population (Oates, 2006:35; Walter, 2004:104). The survey utilizes the questionnaire form to be administered to a sample of participants (Ross, 2005:4). A questionnaire is an instrument used to collect or generate the data required by the study (Ross, 2005:4).

3.3.1.2. The planning and design of a survey involve:

- Data requirements: The decision about which data to gather has to be made premeditatedly. The decision has to be whether the data source is to be directly topic-related or indirectly topic-related. The latter includes age of respondents, gender, size of company, etc. (Oates, 2008:104).
- Data-generation methods: This includes the choice between the methods to collect the data such as questionnaires, interviews, observations and documents (Oates, 2008:104).
- Sampling frame: The sampling frame concerns the collection from a targeted population of people, events, and/or documents from which the sample will be drawn (Oates, 2008:104).
- Sampling technique: The choice of the way the actual people, events or documents are to be selected recognizes the sampling technique. The kinds

of sampling techniques are probability and non-probability. The probability means that the research has a strong conviction that there is a high probability that the sample drawn represents the population being studied. Conversely, non-probability suggests that the researcher is not certain whether the sample represents the overall population (Oates, 2008:104).

- Response rate and non-responses: Getting responses from participants is crucial, therefore, it is suggested that there should be a strategy devised to increase the responses, such that there is emphasis on the purpose of study and the knowledge hoped to achieve (Oates, 2008:104).
- Sample size: The decision about how big the final sample should be needs to be considered since there are some forms of statistical analysis that are not reliable and viable with a small sample (Oates, 2008:104).

3.3.1.3. Evaluating survey-based research

Advantage	Disadvantage
• Provide enough coverage of people or events to represent population (Oates, 2008:104). • Provide enough data in a short time with less cost (Oates, 2008:104). • Associated with quantitative analysis (Oates, 2008:104). • Studies can be replicated (Oates, 2008:104). • Distribution is flexible to suit people with no interpersonal and communication skills (Oates, 2008:104).	• Lack of depth (Oates, 2008:104). • There is a focus on numbers and subjects other than numbers are overlooked (Oates, 2008:104). • Failure to examine incremental process and change (Oates, 2008:104). • Failure to establish cause and effect of realities (Oates, 2008:104). • Lack of face to face contact, thus participants cannot be judged in terms of accuracy and honesty (Oates, 2008:104).

Table 3.3: Advantages and disadvantages of the survey strategy

3.3.2. The application of the survey strategy in this study

To analyse the relationships between the system development methodologies and IT project success, we conducted a survey among information systems and Software systems workforces. A survey obtains the same kinds of data from a large group of people in a standardized and systematic manner (Oates, 2006:35; Walter, 2004:104). The main concern was to draw the pattern and trends from a population group such as determining the most effective critical success factors and system development methodologies in the development of IT projects. An anonymous survey was conducted because the study explored the employees' opinions about the development of information technology project success in their organizations.

The target participants were information systems developers, software developers, team managers or leaders and relevant stakeholders who participated in the development of IT projects. Practically, the target participating organizations were formally addressed to determine whether they would be willing to participate in the study. Followed by sending (through an email), a questionnaire package, including the cover letter and the request note. Due to the positive response hints, a total of 600 questionnaires were distributed to the organizations. Ultimately, after frequent follow-ups, a total of 144 questionnaires were returned, which amounted to a response rate of 24%. However, the completed questionnaires amounted to 132 which means that 12 questionnaires were returned blank. In addition, a reward was offered which stipulated that the participants would also be able to get the outcome of the study upon request.

3.3.3. The reasons for choosing the survey were to:

- Provide enough coverage of people or events to represent the population (Oates, 2008:104); we were able to reach as many people as possible via e-mails.
- Provide enough data in a short time at a lower cost (Oates, 2008:104), we made the participation request via e-mail and telephonic contact, and there was little need for face-to-face contact. Thus, the response was more effective via e-mail which took less time and involved little cost.
- Allow postal and web observation and communication which is suitable to non-interpersonal and communication skills (Oates, 2008:104). Even introverted personnel were able to express their opinions freely.
- Measure the subjects' opinions; they represent a viable approach for investigating the developer's perceptions toward system development methodologies.

- Determine whether a relationship holds for more than one case, not just one case. This study will investigate the perceptions of software developers, IS developers regarding system development methodologies and other critical success factors that contribute to IT project success.

Despite the outlined positive facts, the primary observed problem associated with the survey approach was that we didn't have an adequate approach to address non-respondent information.

3.4. Data-generation methods

3.4.1. Data-generation method employed in this study

In this section we learn about the questionnaire survey method which was chosen from among other data-generation methods to be used in this study. The positivist paradigm embraces the survey strategy and questionnaire generation method (Mertens 2005:2; Oates, 2008:282). Alternatively, other types of data-generation methods exist, namely interviews, observations, and documents (Mertens, 2005:2; Oates, 2008:282). The questionnaire will be discussed extensively based on the following aspects: definition, type, advantage, disadvantage and reliability and validity.

3.4.1.1 Definition

A formal standardized questionnaire is a survey instrument used to generate data from individuals or a group (Kelly *et al.,* 2008:122; Ross, 2005:3). A question is declared a standard when similar sets of questions and the same system of coding responses are used to investigate a target group (Ross, 2005:3). It is a pre-defined set of questions organized in a pre-determined order (Oates, 2008:219). Questionnaires are mostly employed in a large population and where views are difficult to be elicited (Weaver, 2004:168).

3.4.1.2 Types of questionnaires

- Open questions – this type of question allows respondents to answer by providing any type of response that best suits them (Oates, 2008:186; Weaver, 2005:158; Ross, 2005:23; Kelly *et al.,* 2008:123).

- Closed questions – provide a fixed type of question which will allow the respondents to select from a pre-defined set of answers (Oates, 2008:186; Weaver, 2005:158; Ross, 2005:23; *Kelly et al*., 2008:123).
- Contingency questions – accommodate only the sub-group of respondents and are regarded as a special case of closed–ended response (Ross, 2005:23).

3.4.1.3 The advantages of questionnaire

- Questionnaires are more economically feasible (Oates, 2008:186).
- Questionnaires are easy to be completed and analysed (Oates, 2008:186; Ross, 2005:27; Kelly *et al.,* 2008:127).
- Consist of few geographical limitations; moreover, distribution is flexible (Oates, 2008:186).
- Require no social skills of the researcher (Oates, 2008:186).
- Can add new information to increase the current information (Ross, 2005:26)
- Allow the collection of large amount of data in less time (Kelly *et al.,* 2008:126),
- They have response categories that are easy to code (Ross, 2005:27).

3.4.1.4 The disadvantages of questionnaires

- Pre-defined answers can cause frustration (Oates, 2008:186).
- Lack of veracity for the provided answers (Oates, 2008:186).
- No opportunity to correct misunderstandings (Oates, 2008:186).
- Self-administered questionnaires not always appropriate (Oates, 2008:186).
- Open questions may be difficult to answer and analyse (Ross, 2005:26)
- Demand effort and time of respondents (Ross, 2005:27)
- Respondents' writing is likely to be illegible (Ross, 2005:27)

3.4.1.5 Reliability and validity

Validity is the degree to which a question measures what it was initially intended to test (Ross, 2005:27). Reliability embraces the stability of a measure in that the tendency to obtain the same results if the test was to be repeated by using the same subjects under the same conditions (Ross, 2005:27).

3.4.2. The application of questionnaire strategies in the study

We started the research by reviewing the related literature and then we created a survey based on the research questions. A modified questionnaire based on an extensive literature review was employed. Rationally, a questionnaire is a predetermined set of questions assembled in a predetermined order (Oates, 2008:36). This questionnaire was originally administered by Huisman (2000), and it was pre-tested and approved by the Statistics Department. Most of the questions were closed questions.

If evidently a company was suitable for the study it was asked to participate. A questionnaire with a cover letter was sent randomly to the managers who accepted the request, while a few were handed to companies. Henceforth it was upon their agreement to complete the questionnaire voluntarily. In an exceptional case, people were contacted by phone and information concerning the corporate activity of the development of the IT system as per questionnaire was acquired.

In the cover letter the following aspects were stipulated;

- The research topic;
- The purpose of the research;
- The assurance that answers would be kept confidential;
- The emphasis that the completion of the questionnaire was voluntary;
- The results of the study would be provided upon request;
- The deadline;
- The comment; and
- The contact details of the researcher.

The completed questionnaires were sent through e-mail and there were three batches of completed questionnaires returned by courier (those deliveries particularly coming from the provinces such as the Free State, Eastern Cape and Western Cape). As to the rest of the provinces and some of the above-mentioned provinces, the completed questionnaires were received via e-mail except a few that literally had to be fetched from the companies involved.

There were 144 responses from which 132 were completed and hence usable. Objectively, the population was known to be involved in information technology and information system projects. Both IS team manager or leader and IS or software professionals and project team members completed the questionnaire.

The participating province layout was as follows:

1. Please indicate the province in which your organization is based.	
1. Gauteng	1
2. Free State	2
3. North West	3
4. Mpumalanga	4
5. Limpopo	5
6. Kwa Zulu Natal	6
7. Eastern Cape	7
8. Western Cape	8
9. Northern Cape	9
Other, please specify...	10

Table 3.4.2: Province (questionnaire)

To further explore the questionnaire, the questionnaire was categorized with the following constituents:

3.4.2.1 Section A: Demographic information

This section includes the demographic variables. The purpose was to examine the significance of each variable with respect to the relationship between independent and dependent variables. Ideally, in all of the variables, the participants had to choose one of the characteristics and answer all questions with regard to their features. It is important to mention that job description (team manager/leader, system developer and other) was regarded as one of the key factors since different interest groups may have different perceptions in terms of critical success factors affecting IT projects and various experiences using or not using systems development methodologies became essential. The variables consisted of:

- Gender;
- Age;
- Job description;
- Highest qualification obtained;
- Business sector;

- Organization size;
- Province;
- Personal experience;
- Project size; and
- Time of completion.

3.4.2.2 Section B: Critical success factors (CSFs)

This section comprises 49 listed critical success factors. Here the aim was to investigate the impact of the mentioned critical success factors on the success of the IT projects. The participants were requested to disclose their view regarding the underlying factors and furthermore, the degree of influence for each factor was observed. This was achieved by adopting a closed type of question to determine and measure the respondent's views and the extent to which these critical success factors affect the development of IT projects. It was also possible for the participants to fill in the critical success factors which were not mentioned.

A four-point Likert scale was used for most of the questions listed. The scale ranges from "1: To a greater extent" to "4: Not at all". The neutral response, "2 and 3: Neither agree nor disagree", is also provided as an option. The layout of section B is as follows.

To what extent do you agree with the following statements as valid descriptions of the last systems development project you were involved with?	Not at All	To a little extent	To some extent	To a greater extent
1. The project used an appropriate systems development methodology.	1	2	3	4
2. The project had clear defined system goals and objectives.	1	2	3	4
3. The project had a focus on the most important problems or opportunities.	1	2	3	4
4. The project had a simple and straightforward design.	1	2	3	4
5. The project had good training programs for all involved.	1	2	3	4
6. The project had a review schedule after project completion.	1	2	3	4
7. The project had well defined and organized maintenance program.	1	2	3	4
8. The project use an appropriate Hardware and Software technologies.	1	2	3	4
9. The project recognized the level of complexity and risks.	1	2	3	4
10. The project was well-defined.	1	2	3	4
11. During the project users were aware of the requirements	1	2	3	4
12. During the project user's requirements were changed all the times.	1	2	3	4
13. Developers understood user's requirements well.	1	2	3	4
14. There was general agreement on project goals and objectives.	1	2	3	4
15. Different user departments or users had conflicting requirements.	1	2	3	4
16. The project had user's expectations that were real.	1	2	3	4
17. During the project, the developers were familiar with the technology base or infrastructure.	1	2	3	4
18. The project had the developed system which was very complex.	1	2	3	4
	1	2	3	4

19. The projects had intensive user's participation.	1	2	3	4
20. Senior management was highly committed to the project.	1	2	3	4
21. The project had a good project management practices.	1	2	3	4
22. Smaller project milestones.	1	2	3	4
23. The project had a good user-developer trust.	1	2	3	4
24. The project had a good user-developer communication.	1	2	3	4
25. The project had a good user-developer mutual understanding.	1	2	3	4
26. The project had adequate time for development.	1	2	3	4
27. The project had adequate developers available to work on the project.				
28. The project had a good developer's technical expertise.	1	2	3	4
29. The project had a good developer's knowledge of the application domain.	1	2	3	4
30. Developer's knowledge of the systems development process was good.				
31. The project had a good user's knowledge of the systems development process.	1	2	3	4
32. The project used appropriate development techniques.	1	2	3	4
33. The project used appropriate development tools e.g. (project management Or CASE tools.).	1	2	3	4
34. The project applied appropriate systems development methodology knowledge.	1	2	3	4
35. The project had good user training.	1	2	3	4
36. The project had a well-planned installation and adoption of the developed system in user departments.	1	2	3	4
37. The project had the availability of the required technology and expertise to accomplish the specific technical action steps.	1	2	3	4
38. The project had a good act of ''selling'' the final projects to their ultimate intended users.	1	2	3	4
39. The project had timely provision of comprehensive control information at each stage in the implementation process.	1	2	3	4
40. The project had the provision of an appropriate network and necessary data to all key actors.	1	2	3	4
41. The project was able to handle unexpected crises and deviations from plan.	1	2	3	4
42. During the project, the top management was willing to provide the necessary resources and authority/power for project success.	1	2	3	4
43. The project had recruitment, selection and training of the necessary personnel for the team.	1	2	3	4
44. The project covered the entire development life cycle.				
45. The project had efficient coordinating and integrating of activities, documents and personnel.	1	2	3	4
	1	2	3	4
46. The project had regular recording of any activity, mistake, improvement occurred during development phases.	1	2	3	4
47. The project had sufficient testing.	1	2	3	4
48. The project had clear and specified business priorities.	1	2	3	4
49. The project had a clear vision and objectives	1	2	3	4
50. Other, please specify**..**	1	2	3	4
	1	2	3	4
..				

Table 3.4.2.2: List of critical success factors (questionnaire)

3.4.2.3 Section C: The adoption and benefit of systems development methodologies (SDMs)

The focus of this section is the following:

- Commercial systems development methodology used;
- Degree of adoption of systems development methodology;
- Period of systems development methodology;
- Use of systems development methodology;

- Number of people using systems development methodology;
- Measure of system's quality;
- Benefit to follow the process; and
- Future experience.

The other crucial part of the questionnaire was the listing of commercial systems development methodologies. Here the main goal was to learn and measure the most popular SDMs used in the organizations. There was a provision for participants to specify other systems development methodologies that were not listed such as in-house or traditional methods.

A fourpoint Likert scale was used for most of the questions listed. The scale ranges from "1: To a greater extent" to "4: Not at all". The neutral response, "2 and 3: Neither agree nor disagree", is also provided as an option.

1. To what extent is your IS department using the following standard (commercial) system development methods at present? You may mark more than one item.	To a greater	To some extent	To a little extent	Not at All
1.1. Structured Analysis, Design and Implementation of Information Systems (STRADIS).	1	2	3	4
1.2. Yourdon Systems Method (YSM).	1	2	3	4
1.3. Information Engineering (IE).	1	2	3	4
1.4. Rational Unified Process (RUP)	1	2	3	4
1.5. James Martin's RAD (JMRAD)	1	2	3	4
1.6. Extreme Programming (XP)	1	2	3	4
1.7. Other, please specify..	1	2	3	4
....				
..				

Table 3.4.2.4: Types of SDMs (questionnaire)

3.4.2.4 Section D: The reasons for not using SDMs

The focus of this section was to present an understanding of the reasons of the non-use of systems development methodologies in the organizations.

This section contains items that measure the dependent variable, IS developer, team manager or leader's perceptions about the reason why they did not use any system development methodology. Here are the primary questions to test the reason for not using system development methodology. All items are measured on a 1 to 5 Likert scale where 1 represents totally agree and 5 represents totally disagree.

1. To what extent you agree/disagree with the following statements about the last project you were involved with?	Totally Agree	Agree	Neutral	Disagree	Totally Disagree
1.1. The profile of development projects in our IS department doesn't require the use of systems development methods.	1	2	3	4	5
1.2. Systems development methods are too complex or hard to use.	1	2	3	4	5
1.3. The current systems development practice in our IS department is adequate.	1	2	3	4	5
1.4. The experience of the developers in our IS department reduces the need for systems development methods.	1	2	3	4	5
1.5. The benefits of systems development methods use are long-term, whereas costs are incurred short term.	1	2	3	4	5
1.6. There is a lack of experienced staff in our IS department who can effectively use systems development methods.	1	2	3	4	5
1.7. New systems developed with systems development methods are not compatible with legacy systems.	1	2	3	4	5
1.8. Our IS department lacks a suitable environment to support systems development methods.	1	2	3	4	5
1.9. In our IS department there is a lack of management support for the use of systems development methods.	1	2	3	4	5
1.10. The learning curve for systems development methods is very long.	1	2	3	4	5
1.11. The financial investment in systems development methods is too large.	1	2	3	4	5
1.12. In our IS department there is a lot of uncertainty over the benefits of adopting systems development methods.	1	2	3	4	5
1.13. In our IS department there is no clear objectives for adopting systems development methods.	1	2	3	4	5

Table 3.4.2.5: The reasons for not using SDMs (questionnaire)

3.4.2.5 Section E: The project outcomes

In this section, we will learn about the outcomes of IT projects with respect to the use or non-use of systems development methodologies. This will be of most significance for this study in that the usage and effect of systems development methodologies and other critical success factors will be determined. The underlying independent variables are:

- Project outcome;
- The period of use for the specified systems development;
- Processes followed in the development of the project; and
- The results (product) of the project.

3.4.3 The procedure for data collection

The data-collection procedures were executed through the iterative steps shown in Figure 3.4.3. The steps were followed sensitively and persuasively to coerce the target participants to respond appropriately.

- Step 1: Seek through a web search the participants primarily in information systems or software development companies to participate in the research.
- Step 2: Send request via the participating letter to explain the purpose of the study and how organizations and individuals can benefit from the study.
- Step 3: If people indicate that they are willing to participate, the questionnaire providing the instructions about filling in the questionnaire was sent out with contact information was provided and everybody was welcome to ask any question about the content of the questionnaire.
- Step 4: A completed questionnaire was returned.
- Step 4: The follow-up was done to the participants after a week in the form of friendly reminder through sms, e-mail or making of a call.
- Step 5: Upon the receipt of the completed questionnaire, a check was done and the questionnaire was saved in more than one location.
- Step 5: For the questionnaire with various blank spaces, a friendly notification was made to the participants. To some extent the participants would occupy blanks except where they were left blank on purpose. Alternatively, the questionnaire would be saved on the folder to more than one location.
- Step 6: The word of appreciation was sent to the participants who had submitted the completed questionnaire and the reminder concerning the receipt of the outcome upon the request was emphasized.

NB: Points of contacts were e-mail, courier, sms, telephone and personal contact

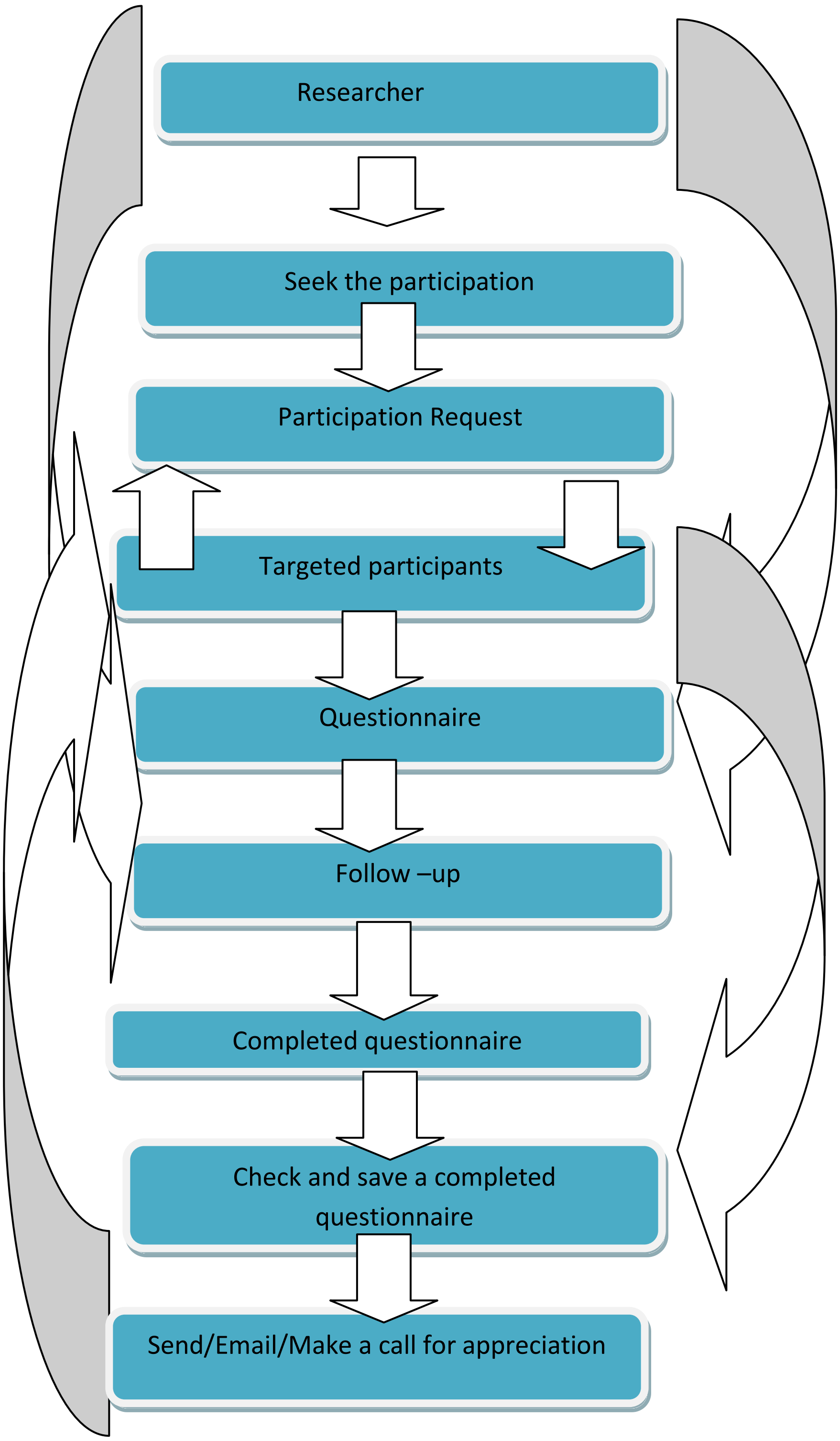


Figure 3.4.3: The data collection procedure for this study

3.4.4 The reasons for choosing a questionnaire data-generation method

Though the survey questionnaire has the stipulated advantages, we still experienced the following challenges:

- The respondents would ignore the deadline.
- To pursue people makes one feel guilty about interfering with their work due to their reluctance to complete the questionnaire.
- There was a possibility that only a small portion of questionnaires would be completely filled in (not all sections).
- The average comment was that the questionnaire was too long (5 pages).
- One sends lot of questionnaires and few were returned.
- People would agree to participate but not even half of them would fulfil the promise.
- Printing of lot of papers and telephone calls is costly, especially when the respondents seem to be positive but eventually do not act appropriately.
- The responses through email exceeded the mail capacity, which needed to be handled strategically.
- It is not always possible to raise a voice on the completed questionnaire.
- To some extent, variables are not filled in appropriately, for example where other SMDs need to be specified, the space would be provided with a number. We recommend the construction of an electronic questionnaire survey.
- One of the reasons from the organizations not to participate in the study was that it was not company policy to disclose their how-about operations or responses.

Overall, though the survey was challenging and gradual; it was very pleasing and enjoyable. Through this study, we got to know people's views and behaviour towards the research. As a result, we realize that a lot has to be done to reveal to people the significance of research in scientific domains.

Furthermore, upon the completion of data attained through the application of research strategy and data generation method, the effort to analyse data to identify trends and patterns occurred. This task can be archived through the data analysis approach (Oates, 2008:38). The subsequent section will give an overview of different types of data analysis used in the research process.

3.5 Data analysis approach

3.5.1. A data analysis approach used in the current study

In terms of data analysis, we used quantitative data analysis to analyse and evaluate the collected data. Quantitative analysis underpins a mathematical and statistical approach to analyse and interpret the data (Oates, 2008:38; Matveev, 2002). The validity of the questionnaire items was assessed using the following: factor analysis, descriptive analysis, correlation between the items, reliability, multiple regression as indicated in the study. The analysis was conducted using Statistica Software Version 10, and the veracity of analysis was checked by Dr Suria Ellis (statistics expert at the North West University).

This survey was superior to statistical methods such as descriptive analysis, frequency table, factor analysis, reliability etc. Table 3.5.1 shows the reliability of the examined construct. The specific constructs on the questionnaire are listed in Table 3.5.1 so as to present the obtained reliability. Most of the items were ranged using a five-point scale ranked from 1) totally agree to 5) totally disagree.

Variables	**Category/No**	**Reference**	**Reliability (Cronbach alpha)**
Gender	Male	Tesch *et al. (2009:660)*	
	Female		
Age	18 or less		
	19 – 25		
	26 – 35		
	36 – 45		
	46 or more		
Qualifications	Senior Certificate	Iivari & Huisman (2007: 41)	
	Diploma		
	University degree		
	Honours degree		
	Master's degree		
	Doctorate degree		
Business sector	Computer Vendor	Iivari & Huisman (2007: 41); Toni (2001:6)	
	Mining		
	Manufacturing		
	Electricity and water		

	supply		
	Consulting		
	Tertiary sector		
	System development		
	Transport and communication		
	Community service		
	Financial Banking		
	Government		
Organizational size	1-50 employees	Huisman (2000); Rainer & Hall (2002:79)	
	51-200 employees		
	201-500 employees		
	501-000 employees		
	>1000 employees		
Experience on SDM	None	Huisman (2000)	
	>1		
	1-2		
	3-5		
	6-10		
	11>		
Project Size	Small	Huisman (2000)	
	Medium		
	Large		
	Very large		
Project duration	< 1yr	Tesch *et al. (2009:660)*	
	1-2yr		
	3-5yr		
	4-more yrs		
Success factors	50	Huisman (2000); Remus (2006:4); Toni (2001:7)	
Commercial SDM	STRADIS		
	YSM		
	IE		
	RUP		
	JMRAD		
	XP		
	Other		
Degree of adoption of SDM	Did not change	Huisman (2000)	
	Major change		
	Major change		
Use of SDM	<1	Huisman and Iivari (2006:44)	
	1-2yr		
	3-5yr		
	4-more yrs		
Intensity use of SDM	None	Huisman and Iivari (2006:44)	
	1-25%		
	26-50%		
	51-70%		
	71% or more		
System quality	8 items	Masrek *et al.*	0.93

		*(*2008:143)	
System process	10 items	Huisman (2000)	0.90
Future experience	6 items	Huisman (2000)	
Reason for non use of SDM	13 Items	Huisman and Iivari (2006:44)	
Project outcome	4 Items	Huisman (2000)	
Project process	9 Items	Huisman (2000)	0.89
Project product	11 items	Huisman (2000)	0.90
Position	Project Manager-Leader	Tesch *et al. (2009:660)*	
	Developer		
	Other		

Table 3.5.1: Data analysis

3.5.2. The reasons for choosing quantitative data analysis were to:

- Provide for scientific measurements (Oates, 2008:263; Krauss, 2005:760; (Matveev, 2002; Mertens, 2005:7).
- Base analyses on well-established techniques (Oates, 2008: 263).
- Obtain data that would allow quantitative predictions (Oates, 2008:263; Johnson & Onwuegbuzie, 2004: 19; Matveev, 2002).
- Permit quick and easy analyses of large volumes of data (Johnson & Onwuegbuzie, 2004: 19; Oates, 2008:263).
- Allow for generalization when replicated on different populations and sub-populations (Johnson & Onwuegbuzie, 2004:19).
- Ensure the reliability of gathered data (Matveev, 2002).
- Observe independent and dependent variables accurately (Matveev, 2002).
- Follow firmly the original set of research goals, arriving at more objective conclusions, testing hypothesis, determining the issues of causality (Matveev, 2002).
- Allowing for longitudinal measures of subsequent performance of research subjects (Matveev, 2002).

Though the quantitative analysis has the above advantages, Matveev (2002) specifies certain weaknesses, which are:

- Failure to provide the researcher with information on the context of the situation where the studied phenomenon occurs;

- Inability to control the environment where the respondents provide the answers to the questions in the survey;
- Limited outcomes to only those outlined in the original research proposal due to closed type questions and the structured format;
- Not encouraging the evolving and continuous investigation of a research phenomenon

The research process for this study was completed by the exploitation of the positivist paradigm, the survey strategy, the questionnaire data-generation method and quantitative data analysis. As we have learned, the common goal of the underlying research methodologies is that they explore the scientific research which investigates the relationships between dependent and independent variables e.g. system development methodology and critical success factor in terms of IT project success.

Summary

In summary, we have learned from previous sections how different paradigms relate to research strategies, data generation methods and data analysis (Table 3.5.2.). The table further presents the examples associated with each research paradigm. Although each of the paradigms has corresponding strategies, data generation methods and data analysis, a researcher may yet adopt a research strategy cutting across research paradigms as per the research questions they desire to answer. In particular, positivism paradigm was regarded as the best suit approach to answer the research question about this study.

Research Paradigm	Strategies	Data-generation Method	Data Analysis	Example
Positivism	Surveys: Experiment Case Study Design and creation	Questionnaire Interviews Observations Documents	Quantitative	Role of Methodology in IT Project Success
Interpretive	Case Study Ethnographies Action Research Design and creation	Interviews Observations Questionnaires Documents	Qualitative	A study of disabled amongst female employees
Critical	Action Research Design and Creation	Interviews Observations Questionnaires Documents	Critical and action-oriented	Absenteeism among grade one learners in public school

Table 3.5.2: Summary for research methodologies

3.6. Conclusion

The objective of this chapter has been to discuss the research methodologies applicable in the development of research projects. In order to understand the background to the research world, a description of research was appended. Henceforth, the roles of the three paradigms which are the positivist, the interpretive and critical approaches were outlined. Furthermore, the participants learned about the different strategies, data generation methods and data analysis approaches that take place in the research process.

It was stated that for this study the positivist paradigm would be adopted in terms of its perceived advantages. The notion behind positivism is that it stands for objectivity, measurability, predictability, controllability and constructs laws and rules of human behaviour (Oates, 2008:286). It is mostly applied in the natural science field where it is claimed to be the most favourable and dominant paradigm researchers employ (Mertens, 2005:7). Furthermore, the Interpretivist paradigm and the Critical paradigm were briefly discussed. Interpretivism addresses social issues; the way people think and act, thus underpins the interpretation of phenomena and making meaning out of this process (Oates, 2008:292; Mertens 2005:11). It is a form of critical research which is concerned with identifying power, conflicts and empowering people; moreover seeking to eliminate them (Oates, 2008:296; Mertens, 2005:15). Typically, the characteristics and the judging of the positivism paradigm have been experiential. Thus, given these facts, it is assumed that Critical and Interpretivist research projects have not been well-established as positive research

(Sale *et al.,* 2002:45; Oates, 2006:305), hence they have been criticized as "non-scientific" (Oates, 2006:305).

Furthermore, the chapter provides a clear outline of the research methodologies followed in this research; it investigates the realities centring on the relationship between systems development methodologies and IT project success, and further articulates the ranking of the most effective critical success factors influencing the IT project success. These approaches have been used to accomplish the research goal; the positivist paradigm, survey, questionnaire and quantitative data analysis were undertaken in a quest to discover knowledge. The underlying approaches embrace the scientific research which is aimed to analyse the observations and measurements. The primary reasons stipulated serve as a support to justify the deployment of the selected research design. Sensibly, the knowledge which was intended to have been discovered (as outlined in the proposal) was believed to be capable of being discovered through these research methodologies. In that, the study accomplished the purpose it was meant for since it adequately addressed the focus of the research questions through the employment of these pre-planned research methodologies.

Chapter 4: Research results

4.1 Introduction

The purpose of this chapter is to present the analysis of the results drawn from the final data collection of this study. The analysis and evaluation process comprises three stages which are: gathering data, analysing data and evaluating the results of the analysis (Weaver, 2004:244). The chapter will take into account the analysis of data and information gained through the research activities. This will be achieved by collating the data in order to identify the trend and patterns and recognize the significance features associated.

The detailed discussion of the results and their implications is presented in Chapter 4. Chapter 5 summarizes the contributions, objectives and limitations of this study and provides avenues for future research.

This chapter is constituted as follows:

- Introduction
- Section A - Demographic information
- Section B – Critical success factors
- Section C – The adoption and benefit of systems development methodologies
- Section D – The reason for not using systems development methodologies
- Section E – The project outcomes
- Section F – Multiple regression
- Conclusion

4.2 Section A - demographic information

The aim of this section is to describe and present the statistical results of demographic information-gathering.

4.2.1 Gender

The respondents were asked to indicate their gender. The number of respondents was 125, where 105 were male and 20 female, thus 84% and 16% respectively. The

7 participants did not respond to this question. It is obvious that most of the respondents were male. The figures are shown in Table 4.1.

Category	Frequency Table: Gender					
	Count	Cumulative Count	Percent of Valid	Cumulative % of Valid	% of all Cases	Cumulative % of All
Male	105	105	84	84	80	80
Female	20	125	16	100	15	95
Missing	7	132	6		5	100

Table 4.1: Gender

4.2.2 Age

In this question, respondents were asked to indicate their age group and the analysis shows that most of respondents were between 25 – 35 by which the category counts to 55 out of 105 respondents; thus giving 44%. Subsequently, age 36 – 45 amounted to 27%, 19-25 gave 23% and 46 or more scaled to 6%. Seven people did not respond. The Figures are displayed in Figure 4.2.

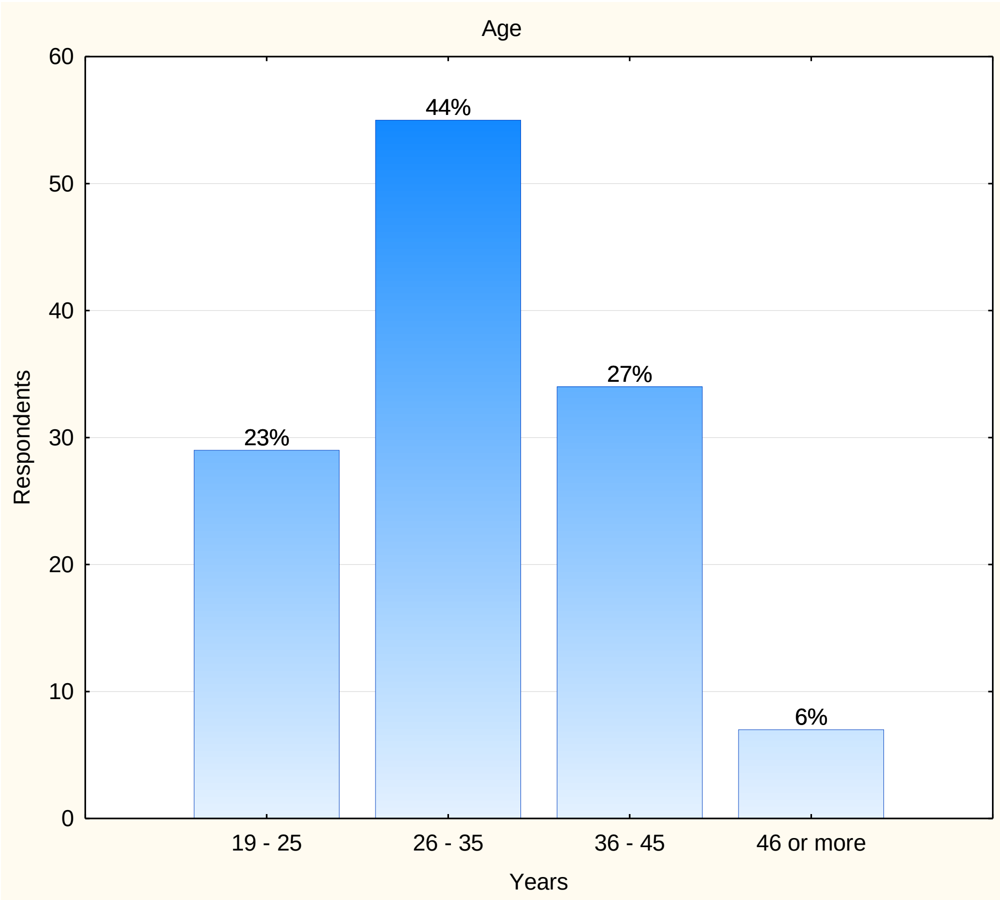


Figure 4.2: Age

In gender versus age; the findings confirm that in the male analysis, respondents accrued a high percentage in the age category. The male respondents were mostly between 25 – 35 years with 42%. In contrast, the female analysis shows a high percentage in the 36 – 45 category which gives 10%. Conversely, the lowest percentage came to be 46 or more for both male and female, with 4% and 2% respectively; however, the 26 – 35 category still confirms a rating of less than 2% for both male and female. The Figures are shown in Figure 4.2a.

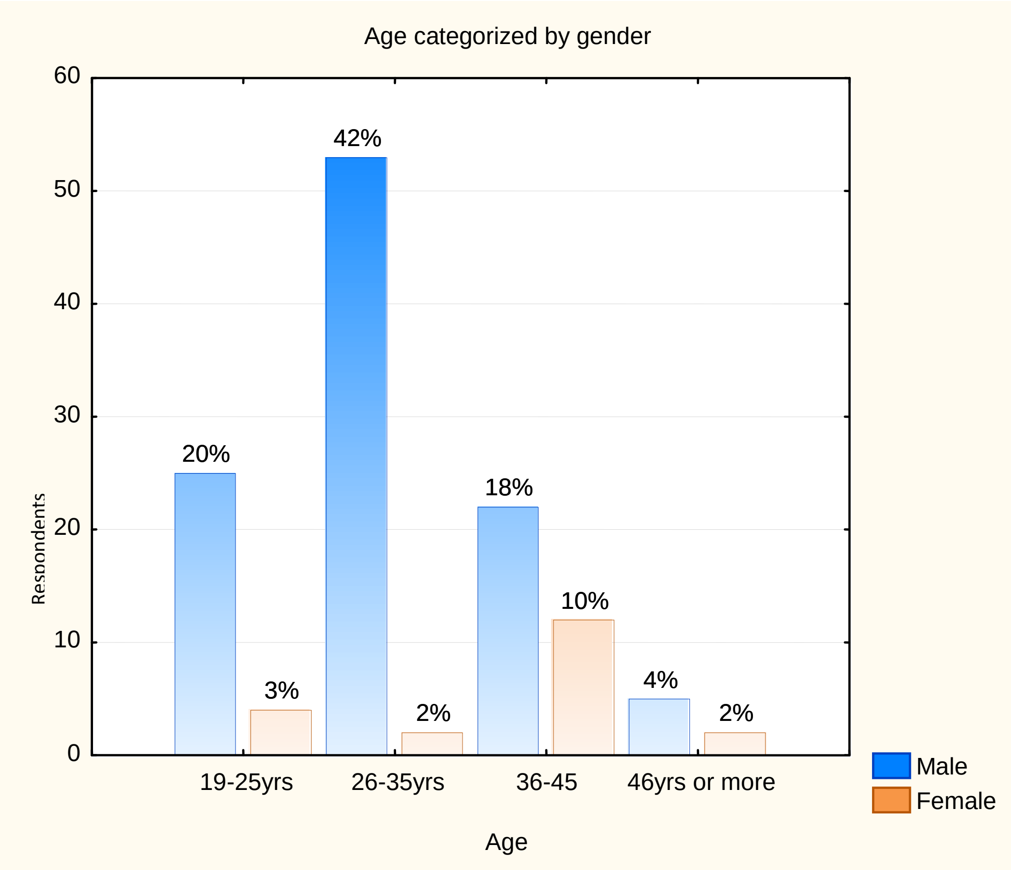


Figure 4.2a: Gender vs. age

4.2.3 Job description

The question was phrased to indicate the job category. The job description is shown in Figure 4.3. The system developers were shown to be 55 out of 123 valid respondents making it 45%. Project manager or project leader was the second highest set of respondents with 35, accounting for 28%. The rest were 'other 'which measured 27%. The number and percentages of job description are shown in Table 4.3 respectively.

Job Category	Frequency Table: Job description					
	Count	Cumulative Count	Percent of Valid	Cumulative % of Valid	% of all Cases	Cumulative %
Project Manager/Leader	35	35	28	28	27	27
Systems developers	55	90	45	73	42	68
Other	33	123	27	100	25	93
Missing	9	132	7		7	100

Table 4.3: Job descriptions

4.2.4 Highest qualifications obtained

The question was asked to provide the highest qualification obtained. Table 4.4 presents the qualifications of the respondents who participated in this study. They are presented in number and percentage respectively. Most of respondents had Honours degrees (32%), with the second most an undergraduate degree (29%), the third highest ranking was a Certificate or Diploma (22%) followed by a masters' degree with 13% and the lowest ranking were a Senior Certificate and a doctoral degree (3% and 2% respectively).

Qualifications	Frequency Table: Qualifications					
	Count	Cumulative Count	% of Valid	Cumulative %of Valid	% of all Cases	Cumulative %
Senior Certificate (High School)	4	4	3	3	3	3
Certificate or Diploma	27	31	22	25	20	23
Undergraduate degree	36	67	29	54	27	51
Honours degree	40	107	32	86	30	81
Masters; degree	16	123	13	98	12	93
Doctoral degree	2	125	2	100	2	95
Missing	7	132	6		5	100

Table 4.4: Highest qualifications of respondents

4.2.5 Business sector

The question was asked to indicate the business sector. The most popular business sector in the study is systems development (36% - 47 participants). Though the rest of the respondents were in various other sectors, they were all involved in systems development projects. Generally, mining and electricity and water supply were less popular businesses reported on in the study. They account for 1% since they have one respondent each. Table 4.5 displays the percentage and number of respondents respectively.

Business sector	Frequency Table: Business sector			
	Count	Cumulative Count	Percent (%) of valid	Cumulative Percent
Computer vendor	7	7	5	5
Mining	1	8	1	6
Manufacturing	7	15	5	11
Electricity & water supply	1	16	1	12
Consulting	6	22	5	17
Tertiary	8	30	6	23
Systems development	47	77	36	58
Transport & communication	16	93	12	70
Financial/Banking	19	112	14	85
Government	4	116	3	88
Other	8	124	6	94
Missing	8	132	6	100

Table 4.5: Business sector

4.2.6 Organizational size

The respondents were asked to indicate the size of their organizations. The size of organization is demonstrated by the number of people employed in it. It is apparent that most of the respondents were working for organizations employing ‘More than 1000 employees; (thus 41% shown in Figure 4.6). Subsequently, the dominant sizes are 1-50 employees and 51-200 employees (37% and 12% respectively). The less popular organizational sizes came to 201-500 employees and 501 – 1000 (5% each).

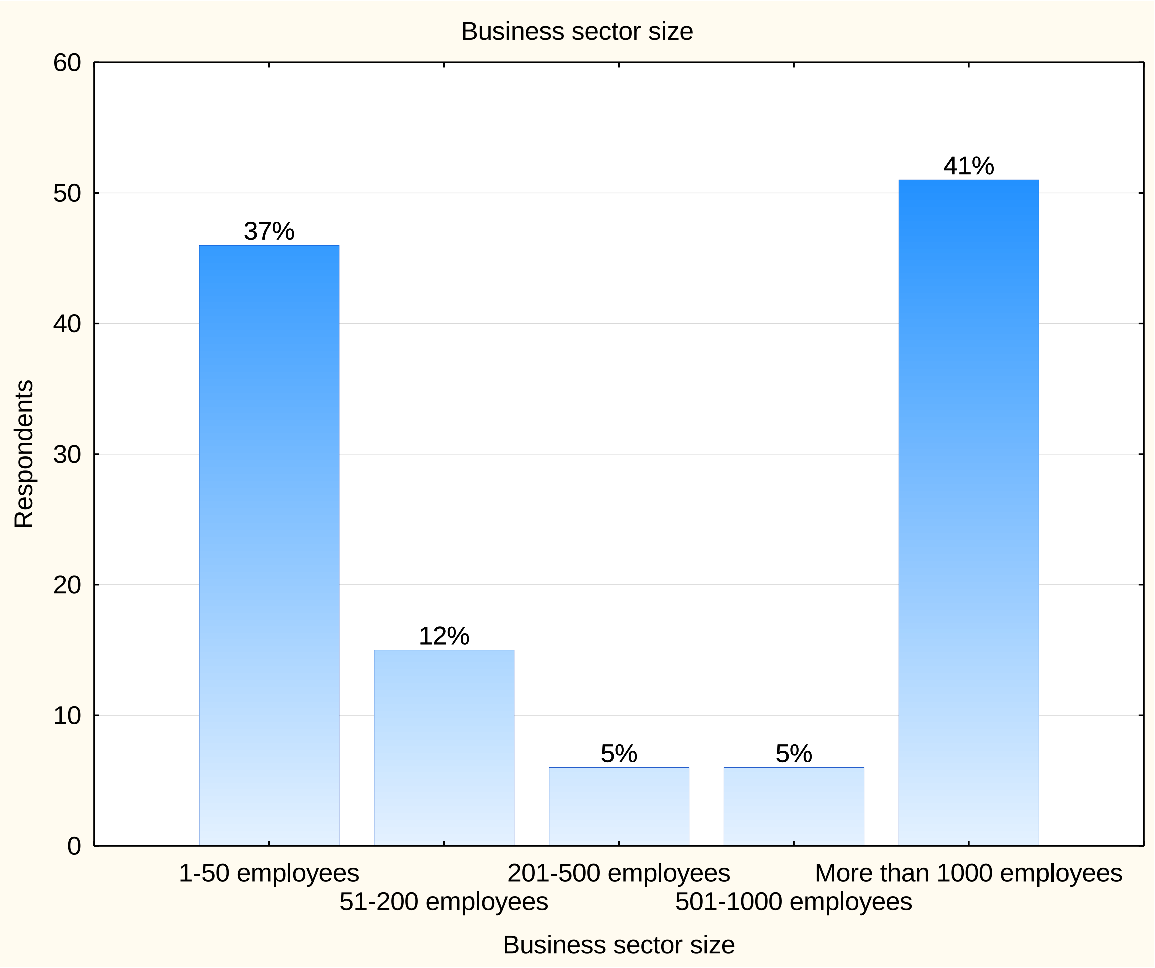


Figure 4.6: Business size

4.2.7 Personal experience

The question was asked of the respondents to reveal their years of experience in the systems development domain. The data analysis showed that the 30% of participants had 11 years or more of experience. The second highest ranking was in the range of 3–5 years which means that participants had between 3 and 5 years of experience in developing systems. Successively, 16 % of the respondents had 6-10 years' experience, and 14% of the respondents had 1-2 years and 3-5 years' experiences. There's evidence of one respondent who did not have any experience in the development of systems (obviously rating at 1% in the study). The percentages are shown in Figure 4.7.

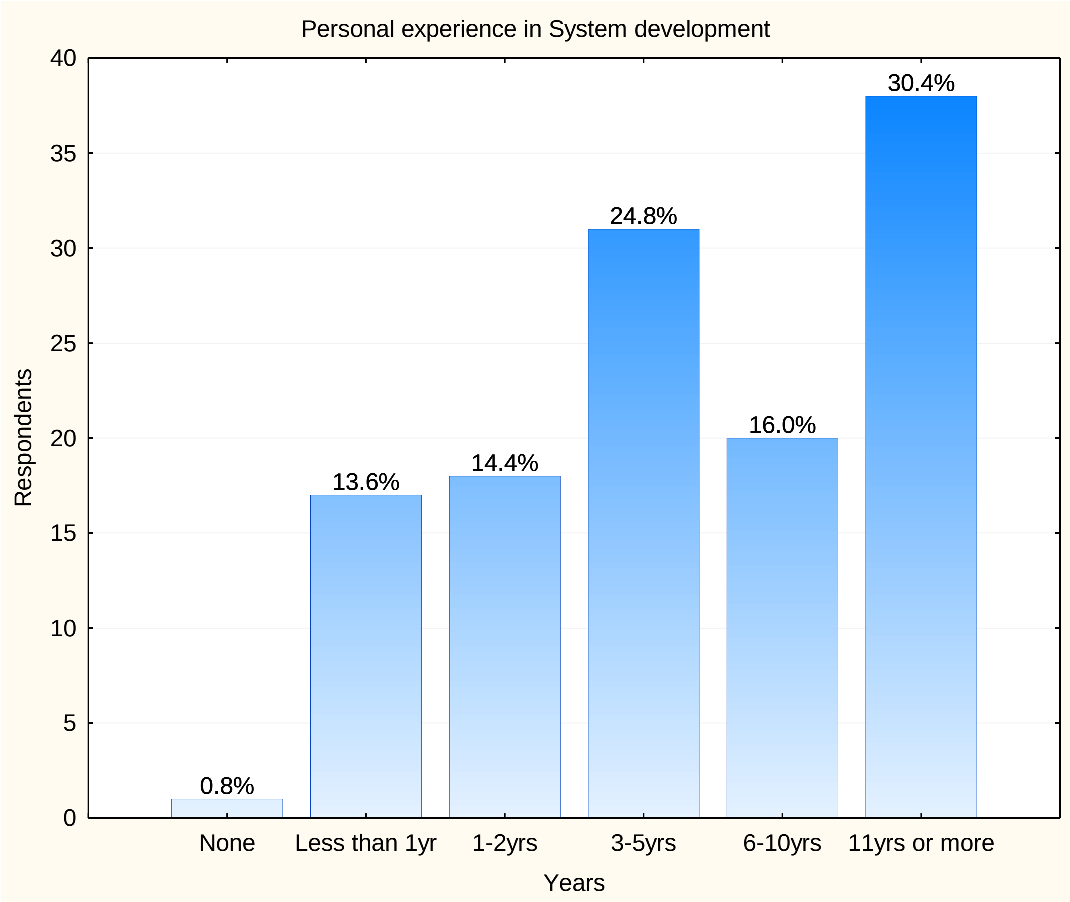


Figure 4.7: Personal experience

This section presents the analysis of biographical information of the respondents that participated in this study. The subsequent section will display the findings of CSFs.

4.3 Section B - Critical success factors

This section is aimed at providing the statistical analysis of critical success factors which are thought to be influencing the IT projects optimistically. In essence, the following examinations were determined: chronological of CSFs as per their means, factor analysis to determine the association of CSFs couple with Cronbach alpha to measure the internal consistency, and finally the observation of the importance of factor analysis according to their means.

The respondents were asked to indicate the critical success factors (CSFs) that had been applied in the last projects the respondents had been involved in and to signify the degree of importance. We measured the mean and assumed the importance of CSFs comprising the highest means (see Table 4.11). This order of ranking is used to observe the most significant critical success factors that influence the success of IT projects. From the analysis, the first critical success factor is assessed to be the "Focus on the most important problems or opportunities" since it constitutes the highest mean of 2.71. Alternatively, 'Recruitment, selection and training of the necessary personnel for the team' gained the lesser mean of 1.67 and it is then observed as the least important critical success factor contributing to the success of the IT projects according to this study. To verify the initial data interpretations, the findings were structured and summarized for reporting purposes in subsequent chapters.

Order of importance	Critical success factor	Critical Success Factor (sort by highest mean)		
		N	Mean	Std. Deviation
1	Focus on the most important problems or opportunities	117	2.71	0.96
2	Senior management Involvement	119	2.69	1.05
3	Use of appropriate Hardware and Software technologies	116	2.54	0.99
4	Good developer's technical expertise	119	2.45	0.91
5	Developers were familiar with the technology base	118	2.34	0.90
6	Developer's knowledge of the systems development process was good	118	2.33	0.93
7	Clear defined system goals and objectives	119	2.31	0.95
8	Good developer's knowledge of the application domain	119	2.26	0.84
9	User's expectations were real	118	2.24	0.84
10	Availability of the required technology and expertise to accomplish the specific technical action steps	119	2.23	0.90
11	Clear and specified business priorities	119	2.22	0.92
12	Clear vision and project plans	117	2.21	0.91
13	Covered the entire development life cycle	117	2.21	0.90
14	Understood user's requirements well	119	2.17	0.92
15	Used appropriate development techniques	118	2.14	0.82
16	Good act of ''selling'' the final projects to their ultimate intended users	119	2.13	1.01

17	Recognized the level of complexity and risks	119	2.12	0.96
18	Top management was willing to provide the necessary resources and authority/power for project success.	119	2.11	0.77
19	Sufficient testing.	119	2.11	0.85
20	Provision of an appropriate network and necessary data to all key actors	119	2.09	0.93
21	Use of an appropriate systems development methodology	119	2.08	0.78
22	There was general agreement on project goals and objectives	119	2.08	0.83
23	During the project users were aware of the requirements	118	2.05	0.83
24	Developed system was very complex.	118	2.05	0.78
25	There was good user-developer trust.	119	2.04	0.86
26	There was a good user-developer communication	115	2.03	0.82
27	Able to handle unexpected crises and deviations from plan	119	2.03	0.81
28	During the project user’s requirements were changed all the times	118	2.02	0.70
29	Well planned installation and adoption of the developed system in user departments.	119	2.01	0.94
30	The project was well-defined	118	1.97	0.76
31	Smaller project milestones	114	1.97	0.86
32	Good user-developer mutual understanding	118	1.96	0.93
33	Adoption of systems development methodology knowledge.	119	1.93	0.77
34	Application of appropriate development techniques e.g. (ERD, DFD).	117	1.93	0.84
35	Good project management practices	118	1.92	0.89
36	Adequate developers available to work on the project	119	1.90	0.75
37	Good user's knowledge of the systems development process and procedures	118	1.90	0.82
38	Regular recording of any activity	113	1.88	0.81
39	Review schedule after project completion	113	1.87	0.81
40	Efficient coordinating and integrating of activities, documents and personnel	119	1.87	0.76
41	Timely provision of comprehensive control information at each stage in the implementation process.	119	1.85	0.70

42	Good user training	119	1.84	0.77
43	Adequate time for development.	117	1.84	0.86
44	Well defined and organized maintenance program	119	1.79	0.72
45	Intensive user's participation	118	1.78	0.84
46	Different user departments or users had conflicting requirements	119	1.74	0.83
47	Simple and straightforward design.	119	1.74	0.80
48	Good training programs for all involved	119	1.71	0.86
49	The project had recruitment, selection and training of the necessary personnel for the team	118	1.67	0.73

Table 4.11: List of critical success factors

4.3.1 Factor analysis and reliability

This subsection presents the factor analyses with the un-rotated factor loadings to evaluate the correspondence of critical success factors. Marked loadings are >.4 to indicate the strength of the link between the factors. Typically, 13 factors were constituted from the related CSFs and were labelled accordingly, along with Cronbach alpha which was analysed to observe the reliability. Most of the factors obtained high Cronbach alpha except F5 and F9. This indicates the good validity of the items. Subsequently, the grouped factors were characterized as the factor number and factor name respectively vs.F1 –Standard methods, F2 - Project's goals and objectives, F3 - User-developer collaboration, F4 - User requirements and involvement, F5 - Technical adequacy and project milestones, F6 - Project plan and control, F7- User training and system deployment, F8 - Developer's proficiency, F9 - Project complexity, F10 - Project development and completion, F11 - Conflicting requirement, F12 - Project resources, F13 - Top management Involvement. See Table 4.11a.

Factor no. & Cronbach alpha	Factor name	Success Factors	Variable	Factor loading
F1 0.87	Standard methods	The project used an appropriate systems development methodology.	B1	.767
		The project used appropriate development tools e.g. (project management Or CASE tools.).	B33	.592
		The project applied appropriate systems development methodology knowledge.	B34	.590
		The project had efficient coordinating and integrating of activities, documents and personnel.	B45	.434
F2 0.82	Project's goals and objectives	There was general agreement on project goals and objectives	B14	-.755
		The project had user's expectations that were real.	B16	-.687
		The project had a clear vision and objectives	B49	-.487
		The project had clear and specified business priorities.	B48	-.444
F3 0.84	User-developer collaboration	The project had a good user-developer mutual understanding	B25	.790
		The project had a good user-developer communication.	B24	.715
		The project had a good user-developer trust.	B23	.688
		The project covered the entire development life cycle.	B44	.544
		Developers understood user's requirements well.	B13	.347
F4 0.66	User requirements and involvement	During the project user's requirements were changed all the times	B12	.706
		The project had a review schedule after project completion	B6	.567
		The projects had intensive user's participation	B19	.552
		The project had good training programs for all involved.	B5	.444
F5 0.50	Technical adequacy and project milestones	The project had the availability of the required technology and expertise to accomplish the specific technical action steps.	B37	.632
		The project use an appropriate Hardware and Software technologies.	B8	.590
		The project had timely provision of comprehensive control information at each stage in the implementation process	B39	-.382
		Smaller Project Milestones	B22	-.342
F6 0.77	Project plan and control	The project was able to handle unexpected crises and deviations from plan	B41	-.714
		The project had regular recording of any activity, mistake, improvement occurred during development phases.	B46	-.454
		The project had recruitment, selection and training of the necessary personnel for the team	B43	-.443
		The project had well defined and organized maintenance program.	B7	-.438
		The project had sufficient testing	B47	-.532
F7	User training	The project had good user training.	B35	.707

0.73	and system deployment	The project had a well planned installation and adoption of the developed system in user departments	B36	.622
		The project had a focus on the most important problems or opportunities	B3	.554
F8 0.84	Developer's expertise	The project had a good developer's technical expertise	B28	.788
		The project had a good developer's knowledge of the application domain.	B29	.716
		The project had adequate developers available to work on the project	B27	.713
		Developer's knowledge of the systems development process was good.	B30	.578
		The project used appropriate development techniques.	B32	.443
		During the project, the developers were familiar with the technology base or infrastructure.	B17	.419
F9 0.46	Project complexity	The project had the developed system which was very complex.	B18	.820
		The project recognized the level of complexity and risks	B9	.455
F10 0.80	Project development and completion	The project had a simple and straightforward design	B4	.782
		The project was well-defined.	B10	.430
		The project had clear defined system goals and objectives.	B2	.417
		The project had a good user's knowledge of the systems development process.	B31	.394
		The project had a good act of ''selling'' the final projects to their ultimate intended users.	B38	.481
F11 0.80	Conflicting requirement	Different user departments or users had conflicting requirements	B15	-.763
F12 0.62	Project resources	The project had adequate time for development.	B26	.913
		The project had the provision of an appropriate network and necessary data to all key actors	B40	.708
F13 0.78	Top management Involvement	Senior management was highly committed to the project.	B20	.711
		During the project, the top management was willing to provide the necessary resources and authority/power for project success	B42	.615
		During the project users were aware of the requirements	B11	.479
		The project had a good project management practices.	B21	.329

Table 4.11a Critical success factors factor analysis

4.3.2 Statistical analysis for factor analysis

This subsection exhibits the significance of the factor analysis in terms of the superior of the mean. In Table 4.11b, the factors are arranged in order of importance. As a consequence, Developer's expertise is observed to be the most favourable factor whereas Conflicting requirement factor seems to be the least favourable factor.

Factor name	Factor no.	Valid N	Mean	Minimum	Maximum	Std.Dev.
Developer's expertise	F8	119	2.45	1.00	3.67	0.61
Project's goals and objectives	F2	119	2.33	1.00	3.50	0.64
Project complexity and risk realization	F9_B9	119	2.32	1.00	4.00	0.77
Top management Involvement	F13	119	2.14	1.00	4.00	0.66
Technical adequacy and project milestones	F5	119	2.13	1.00	3.25	0.52
User-developer collaboration	F3	119	2.12	1.00	3.80	0.62
User training and system deployment	F7	119	2.03	1.00	3.75	0.61
Project complexity System complexity	F9_B18	115	2.00	1.00	4.00	0.82
Standard methods	F1	120	2.00	1.00	4.00	0.80
Project development and completion	F10	120	1.99	1.00	4.00	0.69
Project resources	F12	119	1.96	1.00	4.00	0.73
Project plan and control	F6	119	1.93	1.00	3.75	0.68
User requirements and involvement	F4	119	1.90	1.00	3.50	0.65
Conflicting requirement	F11	119	1.87	1.00	4.00	0.91

Table 4.11b Statistical analysis for factor analysis

In summary, this section displays the statistical analysis of critical success factors to measure their significance in terms of IT project success. It is shown from the analysis that the most significant critical success factor is assumed to be "Focus on the most important problems or opportunities" alternatively, the least critical factor is 'Recruitment, selection and training of the necessary personnel for the team'. The Cronbach alpha was also measured to investigate reliability. Thus, on average they reflect a good consistency since they exhibit high percentages. The next section will discuss the impact of systems development methodology on the success of IT projects.

4.4 Section C - The adoption and benefit of systems development methodologies

The focus of this section is to test the use and the degree of adoption of commercial and in-house systems development methodologies. Furthermore, the analysis measures the quality of the system with respect to the processes or procedures followed in the development of IT systems with the use of SDMs. Consequently, the value of the experience for future use and the reasons for not using of systems development methodologies will be assessed.

4.4.1 Commercial systems development methodology usage

The question asked was to indicate the degree to which SDMs were used in their projects. It is shown that 27% of respondents use STRADIS to a greater extent, 29% to some extent, 13% to a small extent and 32% claimed that they did not use it at all (see Table 4.17a). XP attained the highest percentage (37%) in having been used to some extent, and 15% of respondents have used it to a greater extent. It is also apparent that STRADIS exhibits the highest percentages of use to a greater extent. Alternatively, the analysis shows that 68% of respondents indicated that YSM has not been used at all, whereas 4% claimed to be using YSM to a greater extent. Both STRADIS and XP attained the highest number of respondents (101 and 97 respectively) than other SDMs.

Commercial SDM	Usage Rate in %				Mean	Std Dev	Valid N
	Not at all	To a little extent	To some extent	To a greater extent			
STRADIS	32%	13%	29%	27%	2.50	1.20	101
XP	30%	18%	37%	15%	2.38	1.07	97
IE	43%	17%	27%	13%	2.11	1.11	84
RUP	49%	22%	26%	4%	1.85	0.95	74
JMRAD	51%	27%	13%	9%	1.79	0.98	82
YSM	68%	19%	9%	4%	1.49	0.82	78

Table 4.17a: Commercial SDMs usage rate in %

Furthermore, the respondents were asked to specify their own methods they used other than the one designated. The most popular specified method was Scrum (11% used it to a greater extent, 5% to some extent and 8% did not specify to what degree it was used). The second most popular specified method was Agile, where 11% indicated that they had employed it but without specifying the degree of adoption, 8% had used it to a greater extent and 3% to some extent. The third method which was preferred was the Waterfall Model (5% used it to a greater extent, 3% to some extent whereas 5% did not mention the extent to which they had adopted the method). The rest of the respondents employed methods such as APM, AUP,DSDM, Feature Driven development, Kanban, PMBOK, PRINCE2, SADM, SDLC, SQA, Test Driven development, and VBM Vision-based methodology (each obtained 3%). The percentages and the count of respondents are displayed in Table 4.17b.

Traditional SDM	N=132	Usage Rate in %				
	Frequency	Not at all	To a little extent	To some extent	To a greater extent	Used but degree not specified
AGILE	8			3%	8%	11%
APM	1					3%
AUP	1			3%		
DSDM	1					3%
FEATURE DRIVEN DEVELOPMENT	2			3%		3%
KANBAN	1				3%	
PMBOK	3					8%
SADM	1			3%		3%
SCRUM	9			5%	11%	8%
SDLC	2				4%	
SOA	1				3%	
TEST-DRIVEN DEVELOPMENT	1					3%
VBM-VISION BASED METHODOLOGY	1		3%			
WATERFALL MODEL	5			3%	5%	5%
Missing	94					

Table 4.17b: Custom SDMs usage rate in %

4.4.2 Degree of adaption of systems development methodology

The question was phrased to point out the commercial SDMs and the degree to which they were changed to fit the specific needs of organizations. It can be observed that in STRADIS, 47% of respondents affirm that the listed systems development methodologies were adopted but had not been changed, 35% were adopted but had undergone minor changes whereas 18% were adopted with major changes (see Table 4.18a). The most preferred adopted SDMs were STRADIS and XP, both with 77 respondents.

Commercial SDM	Commercial SDM adaption degree in %			Mean	Std Dev	Valid N
	Did not change	Minor change	Major change			
STRADIS	47%	35%	18%	1.71	0.76	77
XP	52%	34%	14%	1.62	0.73	77
IE	59%	28%	13%	1.54	0.84	54
RUP	65%	26%	9%	1.44	0.79	54
JMRAD	66%	29%	5%	1.42	0.66	50
YSM	66%	26%	8%	1.39	0.59	59

Table 4.18a: Commercial SDMs adaption degree in %

Furthermore, the question was aimed at specifying the other SDMs used and the signification of their adaption. The analyses verify that on average, out of 8 respondents, Agile (13%) had undergone changes but did not specify the degree of change, 13% had undergone minor changes and 9% had performed major changes. Consequently, on Scrum, out of 8 respondents changes had been performed in which 17% did not specify the degree of changes and 17% had performed minor changes. The rest of specified methods in Table 4.18 b, each had one respondent where they had undertaken the changes as per detailed below.

Custom SDM	N =132	Customary SDM adoption degree in %			
	Frequency	Did not change	Minor change	Major change	Adapted but degree Not specified
Agile	8		13%	9%	13%
AUP	1		4%		
In-house method	1				4%
Kanban	1	4%			
PMBOK	8				9%
Scrum	8		17%		17%
Test driven Development	1	4%			

Waterfall Model	1	4%			
Missing	109				

Table 4.18 b: Custom methods adaption degree in %

4.4.3 Period of systems development methodology usage

The question was asked in the questionnaire to indicate how long the SDMs had been in use in their Information systems departments. Out of 115 respondents, 29% indicated that they had been using systems development methodology for 3 to 5 years in the organization. Subsequently, 22% employed systems development methodology for 1 to 2 years, 17% used the methodologies for 6 to 10 years, 8% for less than a year; whereas 13% of respondents confirmed that they did not know (see Figure 4.19).

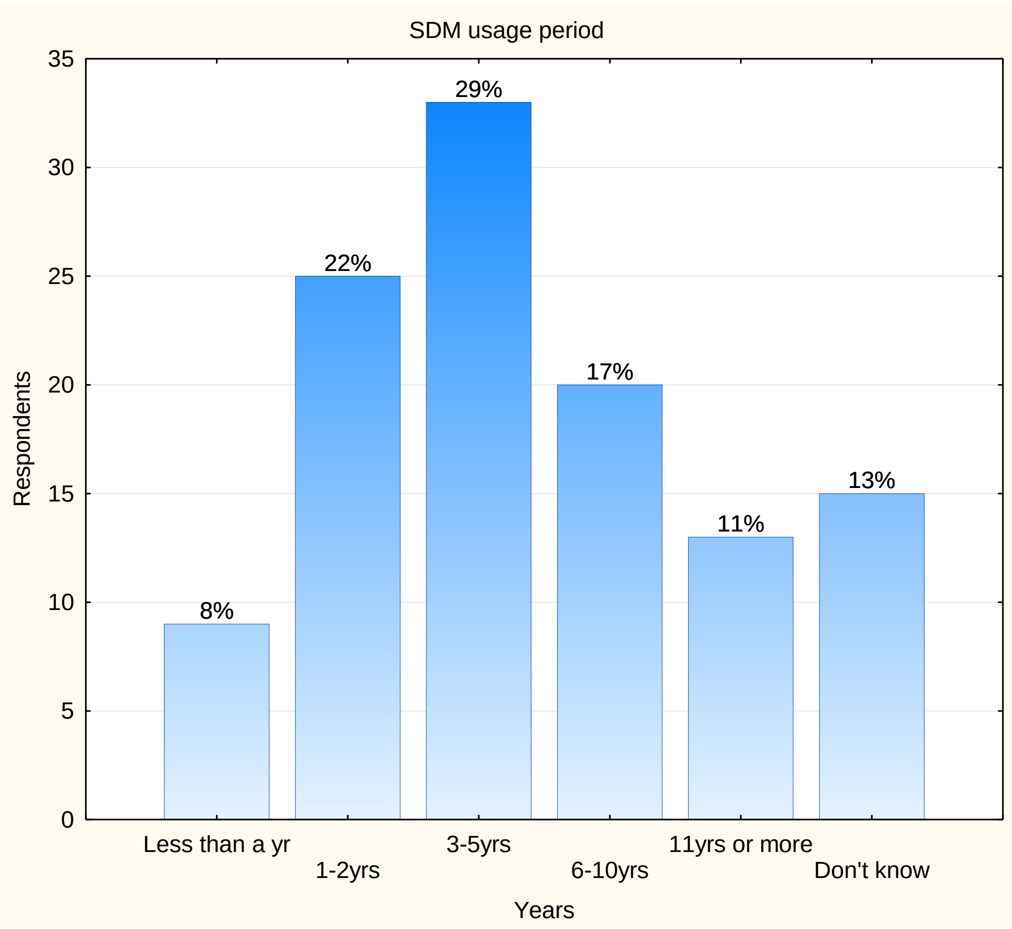


Figure 4.19: Systems development era in %

4.4.4 The proportion of projects developed by using systems development methodologies

The relevant question in the questionnaire, as discussed elsewhere, asked respondents to indicate the proportion of projects that had been developed by applying SDMs. It was observed that 76% or more were confirmed high (with 35% of respondents out of 117 respondents). 18% of respondents answered that 51%-75% and 26%-50% of projects had used systems development methodologies. 17% of respondents claimed that 1%-25% of projects did use systems development methodologies. 12% of respondents affirmed that none of the projects in their organizations had applied any of systems development methodologies (see Figure 4.20).

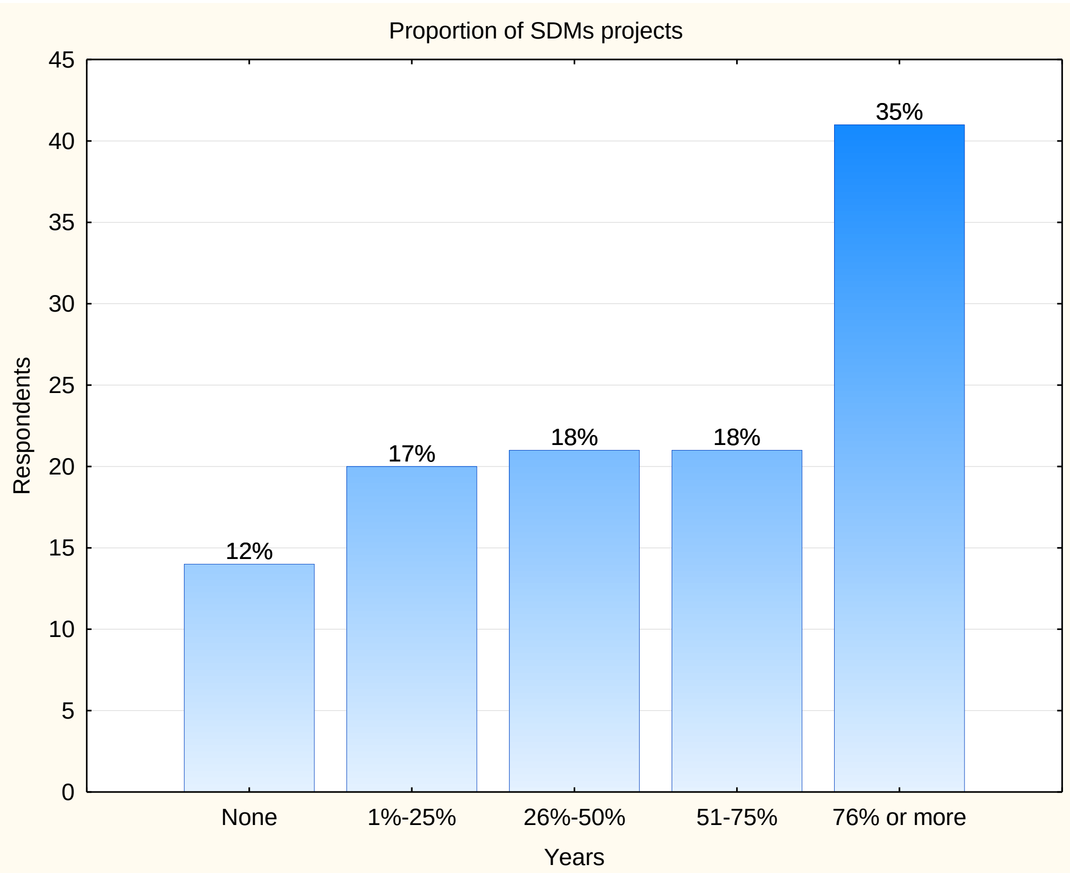


Figure 4.20: Proportion of projects using SDMs

4.4.5 Proportion of people who used systems development methodology

In this question phrased in the questionnaire, the relative numbers of people who have employed systems development methodologies were measured. The analysis verified that 116 participants complied with this criterion, of which 27% respondents claimed that the people who were using systems development methodology could come to 76% or more in the organization. Furthermore, 22% of respondents claimed that 1% to 25% and 51% to 75% of people utilized systems development methodologies in their organizations (refer to Figure 4.21).

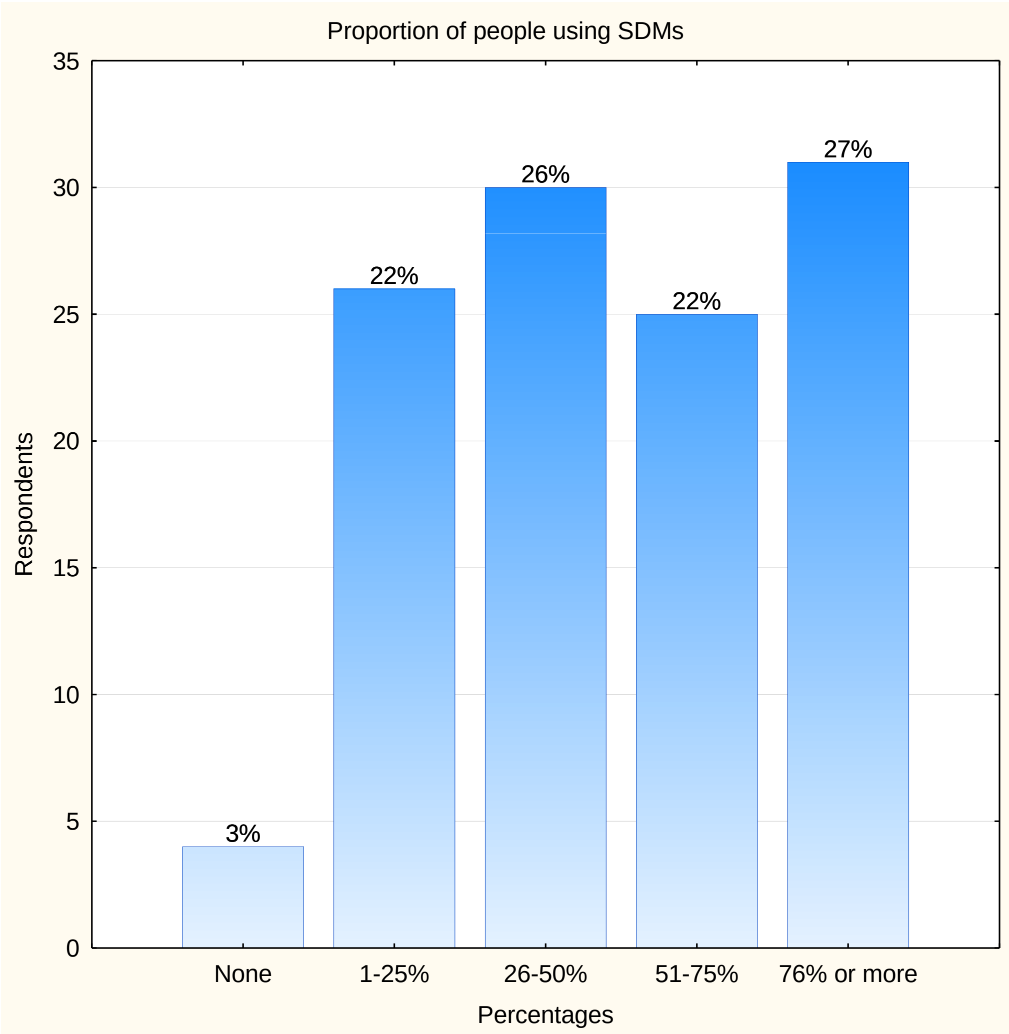


Figure 4.21: Proportion of people using SDMs

4.4.6 Horizontal use of systems development methodologies

The aim of this section is to determine the horizontal uses of SDMs by taking into consideration the proportions of SDMs projects and people employing SDMs. The analysis verified that 34% respondents agreed on the use of 76% or more, 23% respondents confirmed the usage of 51% to 75% , 22% people observed on 26%-50%, 17% respondents claimed the usage at the rate of 1% - 25% usage, and 3%

respondents agreed that SDMs have never been applied. Overall, the average horizontal use of systems development methodologies is 3.47 (refer to Figure 4.22).

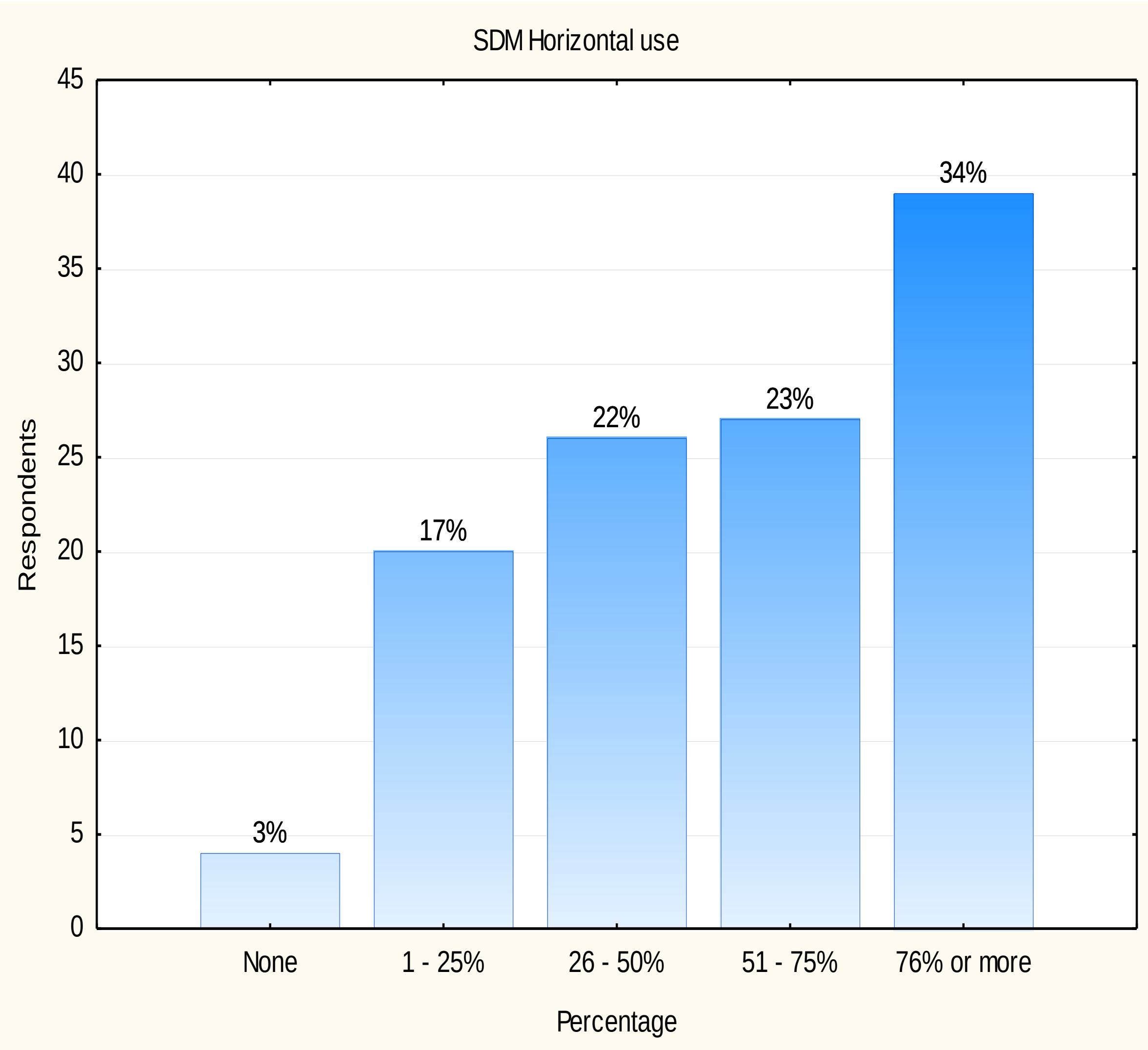


Figure 4.22: Horizontal use of SDMs

4.4.7 The evaluation of systems quality influenced by systems development methodology practices

The question was intended to indicate the impact of systems development on the quality of systems. It is evident from Table 4.22a that most of the respondents agree that using systems development contributes to the development of IT systems. Far fewer respondents totally disagree about the affectivity of SDMs in projects. In particular, the factors are arranged in order from the lowest means; and the assumption is that the lower the mean, the higher the value of the factor.

Impact of SDM on quality of the system	Totally Agree	Agree	Neutral	Dis-agree	Totally Disagree	Mean	Std Dev
Helps to develop better systems.	35%	43%	15%	6%	1%	1.96	0.91
Helps to develop more usable systems.	25%	47%	23%	4%	2%	2.1	0.878
Helps to develop more maintainable systems.	30%	38%	21%	9%	1%	2.12	0.97
Helps to develop more reliable systems.	23%	50%	19%	7%	1%	2.13	0.89
Helps to develop more functional systems.	23%	50%	18%	5%	4%	2.16	0.96
Helps to make users more satisfied with our systems.	23%	46%	22%	6%	2%	2.17	0.92
Helps to develop more efficient systems.	23%	40%	30%	5%	1%	2.21	0.89
Helps to develop more portable systems.	17%	30%	41%	7%	5%	2.52	1.00

Table 4.22a The impact of SDMs on quality of the system

4.4.7.1 Factor analysis and reliability

Table 4.22b represents the factor analysis; un-rotated factor loadings evaluate the correspondence of factors that influence the quality of the system. Marked loadings are >.400000 to indicate the strength of the link between factors. In the table from which the use of systems development methodologies was quantified, all items emphasize the impact of systems development methodologies in the development system they are and compounded on FI - which is named systems development method impact. Furthermore, Cronbach alpha was measured to determine the reliability of factors. The reliability is shown to be high (0.93), which thus proves that the validity of the items is good.

Factor & Cronbach alpha	Factor Name	Impact of SDM on quality of the system	Variables	Factor loading
F1 0.93	Systems development method impact	Helps to develop more reliable systems	C6.2	0.89
		Helps to develop more efficient systems	C6.5	0.86
		Helps to develop more functional systems	C6.1	0.86
		Helps to develop better systems	C6.7	0.84
		Helps to make users more satisfied with our systems	C6.8	0.83
		Helps to develop more maintainable systems	C6.3	0.81
		Helps to develop more usable systems	C6.6	0.79
		Our systems development methodology helps to develop more portable systems.	C6.4	0.77

Table 4.22b: Factor analysis and reliability (impact of SDMs on system quality)

4.4.8 The evaluation of systems development process quality influenced by systems development methodology practices

This section addresses the question wherein the respondents were asked to assume the benefit of adopting systems development methodologies in the development processes of IT systems. It was statistically significant that most of the respondents out of 112 agreed that executing the processes of systems development methodologies was an advantage in terms of the processes followed in the development of IT systems (Table 4.24a). Very few respondents had totally disagreed. In particular, the aspects in Table 4.24a were measured and arranged in the order of their means (from lowest to highest) from which one can assume the value of their importance. The most favourable aspect indicates that SDMs help to improve the quality of the systems.

Process factors influenced by SDMs	Totally Agree	Agree	Neutral	Dis-agree	Totally Disagree	Mean	Std Dev
Helps to improve the quality of the systems.	23%	55%	16%	6%	0%	2.05	0.80
Helps to improve our IS department's reputation of excellent work.	23%	52%	19%	5%	2%	2.10	0.87
Helps to achieve the goals of our IS department.	13%	65%	17%	5%	0%	2.13	0.69
Helps to improve the functionality of new applications.	14%	59%	19%	8%	0%	2.21	0.79
Helps to increase the productivity of the application developers.	17%	49%	29%	5%	0%	2.23	0.79
Helps to develop new applications faster.	17%	53%	21%	8%	1%	2.23	0.86
Helps to decrease the cost of systems maintenance	18%	47%	21%	13%	0%	2.30	0.92
Improves the morale in my IS department.	17%	38%	31%	12%	3%	2.46	0.99
Helps to develop the cost of systems development.	13%	39%	34%	10%	4%	2.51	0.97
Helps to improve the documentation of the systems.	16%	38%	25%	18%	3%	2.53	1.05

Table 4.24a: SDMs factors influencing the SD process (percentage and mean)

4.4.8.1 Factor analysis and reliability

Table 4.24b presents factor analysis using an un-rotated factor loading to evaluate the correspondence between SDMs factors that influence the processes followed in the development of IT systems. Marked loadings are >.400 000 to indicate the strength of the link between the factors. F1 is made up of seven items that include the productivity of the developed system, so it is named process production and departmental development. Similarly, F2 is made up of three items that involve the process quality of the developed system hence it is called process quality improvement and cost reduction. Furthermore, Cronbach alpha was measured to determine reliability. The reliabilities of the underlying factors are shown to be high (0.88 and 0.82), thus proving that the validity of the items is good.

Factor & Cronbach alpha	Factor Name	Process factors influenced by SDMs	Variables	Factor loading
F1- 0.88	Process production and departmental development	Helps to develop new applications faster	C7.1	.945
		Helps to improve the functionality of new applications.	C7.2	.918
		Helps to increase the productivity of the application developers	C7.3	.767
		Helps to decrease the cost of systems development	C7.4	.563
		Helps to achieve the goals of our IS department	C7.9	.542
		Helps to improve our IS department's reputation of excellent work.	C7.10	.502
		Improves the morale in my IS department.	C7.8	.459
F2- 0.82	Process quality improvement and cost reduction	Helps to improve the documentation of the systems.	C7.7	.903
		Helps to decrease the cost of systems maintenance	C7.6	.841
		Helps to improve the quality of the systems	C7.5	.773

Table 4.24b: Process factors factor analysis and reliability

4.4.9 Future experience

The closed question was posed in the questionnaire to specify the future use of systems development methodologies in the coming two years. Generally, the analysis confirms that most of the respondents were neutral as to whether they were going to use SDM in the next two years. The second highest ranking is the “Agree” followed by “Disagree”, “Totally Disagree” and “Totally Agree”. Typically, the aspects in Table 4.30 were measured; and it was found that the most preferred aspect indicated making more use of SDMs. The following aspects are arranged in order of importance with respect to the lowest mean.

Factors pertinent in the issue of future use of SDMs	Totally Agree	Agree	Neutral	Disagree	Totally Disagree	Mean	Std Dev
Make more use of our SDMs.	29%	46%	18%	4%	3%	2.05	0.94
Supplement SDMs with other methodologies.	13%	38%	36%	8%	5%	2.56	1.00
No change.	10%	14%	43%	21%	13%	3.13	0.78
Replace SDMs.	11%	14%	35%	34%	7%	3.13	1.09
Abandon the use of SDMs	5%	8%	20%	24%	44%	3.95	1.11
Other	11%	0%	78%	11%	0%	0%	1.17

Table 4.30: Project future experience of SDMs %

In wrapping up, this section just dealt with presents the degree of adoption of commercial and in-house systems development methodologies. The next section will outline the reasons for not using SDMs in the development of IT systems; thus the experiences with the employment of SDMs will be measured.

4.5 Section D - the reasons for not using systems development methodologies

This section is aimed at determining the intensity in the context of not using systems development methodologies in the development of IT systems. Thus, the reasons for not adopting any of systems development methodologies will be investigated.

4.5.1 Experience with the interaction of Systems Development Methods

The question in the questionnaire about interaction was intended to capture a sense of the experience that led to the non-use of SDMs. The analysis shows that only 41 respondents answered this question in which 12 respondents confirmed that they have never used any commercial or in-house methods, 19 respondents considered using systems development methodologies but decided not to use it and 10 respondents claimed that they have used it but abandoned it (refer to Figure 4.31, Table 4.31).

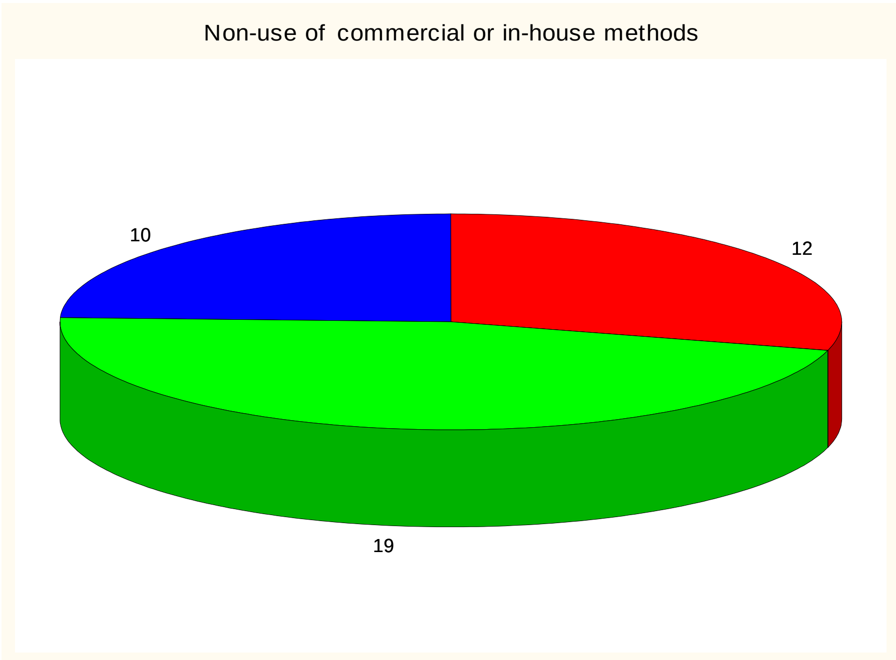


Figure 4.31: Experience of the non-use of SDMs

Legend for the chart above

Never used

Considered use but decided otherwise

Used but abandoned

Category	Non-use of commercial SDMs				
	Count	Cumulative Count	Percent (%) of Valid	Cumulative % of Valid	% of all Cases
1 - Never Used	12	12	29.27	29.27	9.09
2 - Consider used but decided otherwise	19	31	46.34	75.61	14.39
3 - Used but abandoned	10	41	24.39	100.00	7.58
Missing	91	132	68.94		68.94

Table 4.31: Experience of the non-use of SDMs %

4.5.2 The reasons for not using a systems development methodology

This question in the questionnaire required that the respondents should state the reasons for not using SDMs in the last project they had been involved in. The popular reason why systems development methodologies were not used is the one with the lowest mean in Table 4.32 which states that benefits of SDM use are long-term, whereas costs are incurred short-term, and the least mentioned aspect reflects that the profile of development projects in our IS department does not require the use of SDMs.

Reasons for the non-use of SDMs	Totally Agree	Agree	Neutral	Disagree	Totally Disagree	Mean	Std Dev
The benefits of SDMs use are long-term, whereas costs are incurred short term.	14%	47%	29%	9%	2%	2.38	0.89
There is a lack of experienced staff in our IS department who can effectively use SDMs.	19%	24%	35%	17%	5%	2.67	1.13
Our IS department lacks a suitable environment to support SDMs	5%	33%	24%	36%	2%	2.97	0.99
The current systems development practice in our IS department is adequate.	9%	33%	17%	33%	9%	3.00	1.23
The experience of the developers in our IS department reduces the need for SDMs	12%	28%	19%	31%	10%	3.00	1.17
The learning curve for SDMs is very long.	4%	33%	26%	32%	5%	3.02	1.01
In our IS department there is a lack of management support for the use of SDMs	5%	33%	24%	36%	2%	3.09	1.14
In our IS department there is no clear objectives for adopting SDMs	4%	26%	30%	35%	5%	3.1	0.89
New systems developed with SDMs are not compatible with legacy systems.	3%	21%	41%	31%	3%	3.1	0.92
In our IS department there is a lot of uncertainty over the benefits of adopting SDMs	5%	19%	37%	37%	2%	3.12	1.04

The financial investment in SDMs is too large.	7%	21%	30%	37%	5%	3.12	0.98
SDMs are too complex or hard to use.	5%	17%	29%	41%	7%	3.28	1.01
The profile of development projects in our IS department doesn't require the use of SDMs	5%	29%	12%	29%	24%	3.38	1.28

Table 4.32: Reasons for not using SDMs %

4.5.2.1 Factor analysis and reliability

Table 4.33 represents the factor analysis; with the use of un-rotated factor loadings to evaluate the correspondence of factors which describe the reason for not using systems development methodologies. F1 is formulated from the items that describe the failure to adopt SDMs in the development of IT systems; therefore the factor is named resource deficiency and benefit uncertainty. F2 characterizes the items that explain the existence method competence and the cost accumulation in SDM practices. F3 categorizes the "Ambiguity and irrelevancy of SDMs". Furthermore, the Cronbach alpha was measured to determine the reliability. The reliabilities of factors are shown to be good (0.87, 0.68 and 0.75), thus proving that the validity of the items is good.

Factor & Cronbach alpha	Factor name	Reasons for not using SDMs	Variables	Factor loading
F1- 0.87	Resource deficiency and benefit uncertainty	The learning curve for systems development methods is very steep.	D2.10	0.82
		In our IS department there is a lack of management support for the use of systems development methods	D2.9	0.77
		In our IS department there is no clear objectives for adopting systems development methods	D2.13	0.77
		The financial investment in systems development methods is too large	D2.11	0.75
		In our IS department there is a lot of uncertainty over the benefits of adopting systems development methods.	D2.12	0.71
		Our IS department lacks a suitable environment to support systems development methods.	D2.8	0.51
		New systems developed with systems development methods are not compatible with legacy systems	D2.7	0.51

F2- 0.68	Existence method competency and cost accumulation	The benefits of systems development methods use are long-term, whereas costs are incurred short term.	D2.5	0.88
		The current systems development practice in our IS department is adequate.	D2.3	0.71
F3- 0.75	Ambiguity and irrelevancy of SDMs	The profile of development projects in our IS department doesn't require the use of systems development methods	D2.1	0.92
		The experience of the developers in our IS department reduces the need for systems development methods	D2.4	0.63
		Systems development methods are too complex or hard to use	D2.2	0.56
		There is a lack of experienced staff in our IS department who can effectively use systems development methods	D2.6	0.41

Table 4.33: Reasons for not using SDMs factor analysis and reliability

4.5.2.2 Statistical analysis of the impact of SDMs

This section is aimed at presenting the statistical analysis of the factor analysis components vs. SDM on quality of the system (Table 4.22b), process factors influenced by SDMs (Table 4.24b), and reasons for not using SDMs (Table 4.33) with regard to the impact of SDMs in the development of IT systems. The factors are arranged in order of importance of their means. It is assumed that the lower the mean, the greater the significance of the factor. See Table 4.34.

Factors	Valid N	Mean	Minimum	Maximum	Std.Dev.
Systems development method impact	112	2.17	1.00	5.00	0.77
Process production and departmental development	112	2.26	1.00	4.00	0.65
Process quality and cost management	112	2.29	1.00	4.33	0.79
Resource deficiency and benefit uncertainty	58	3.07	1.00	4.86	0.75
Existence method competency and cost accumulation	58	2.69	1.00	5.00	0.91
Ambiguity and irrelevancy of SDMs	58	3.08	1.00	5.00	0.88

Table 4.34: Statistical analysis of the impact of SDMs

In conclusion, this section presented the impact of SDMs on the development of IT systems; thus the most viable SDMs were measured. The interest in continuing to use the SDMs was also evaluated. Henceforth, the reasons for not using SDMs were also observed. The next section will outline the outcome of the use of systems development methodologies.

4.6 SECTION E - The project outcomes

4.6.1 Project size

The question from the questionnaire was phrased to elicit the size of the last project that the respondents had been involved in. It emerged that large projects were most popular (40%), followed by medium, very large and small (30%, 16% and 10% respectively). Some (3%) had not been involved in any information systems development. The percentages and counts are displayed in Figure 4.9.

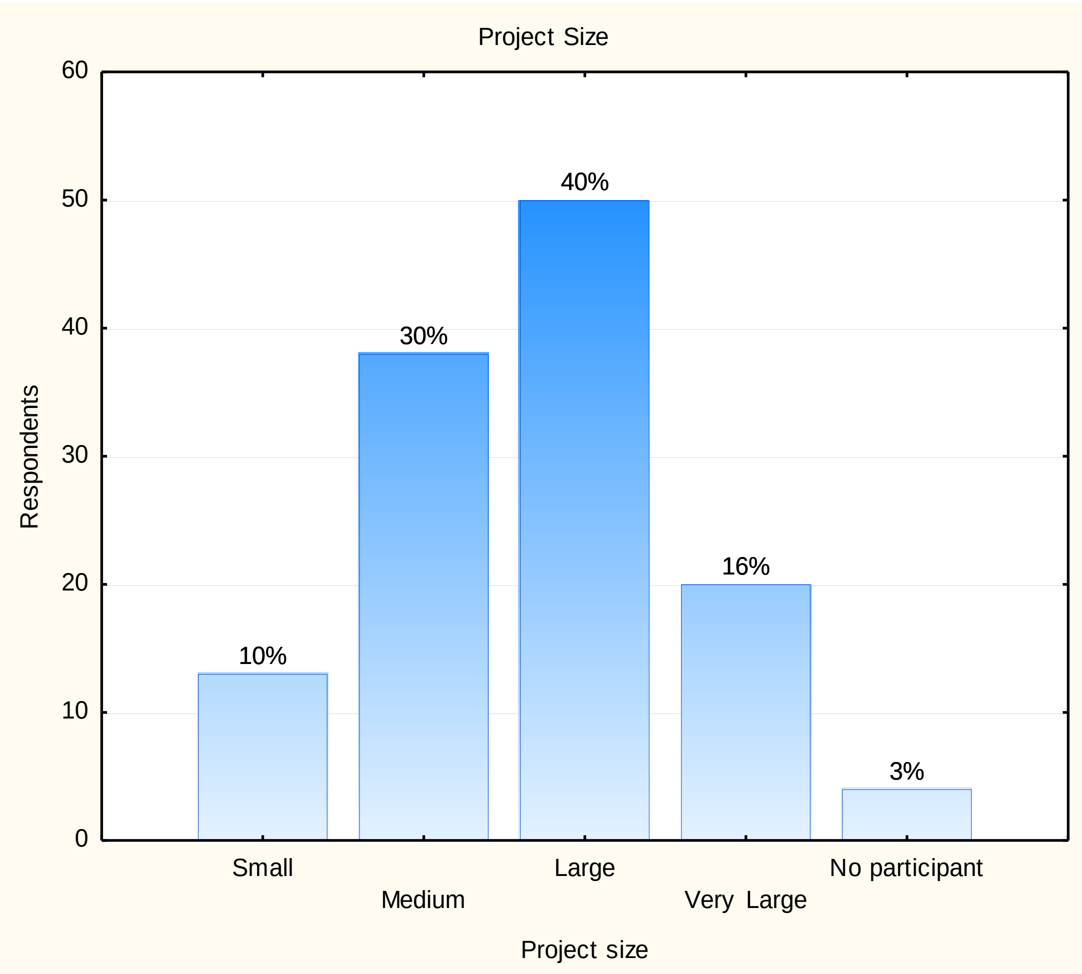


Figure 4.9: Project size

4.6.2 Systems development project completion time

The respondents were asked from the questionnaire to indicate how long it took to complete the last IT project they had been involved in. Figure 4.10 shows that most of the respondents (53%) were able to complete the project within a year. Subsequently, the rest of the respondents who answered this question were able to complete in 1-2 years, 3-5 years and 6 years or more (34%, 13% and 1% respectively).

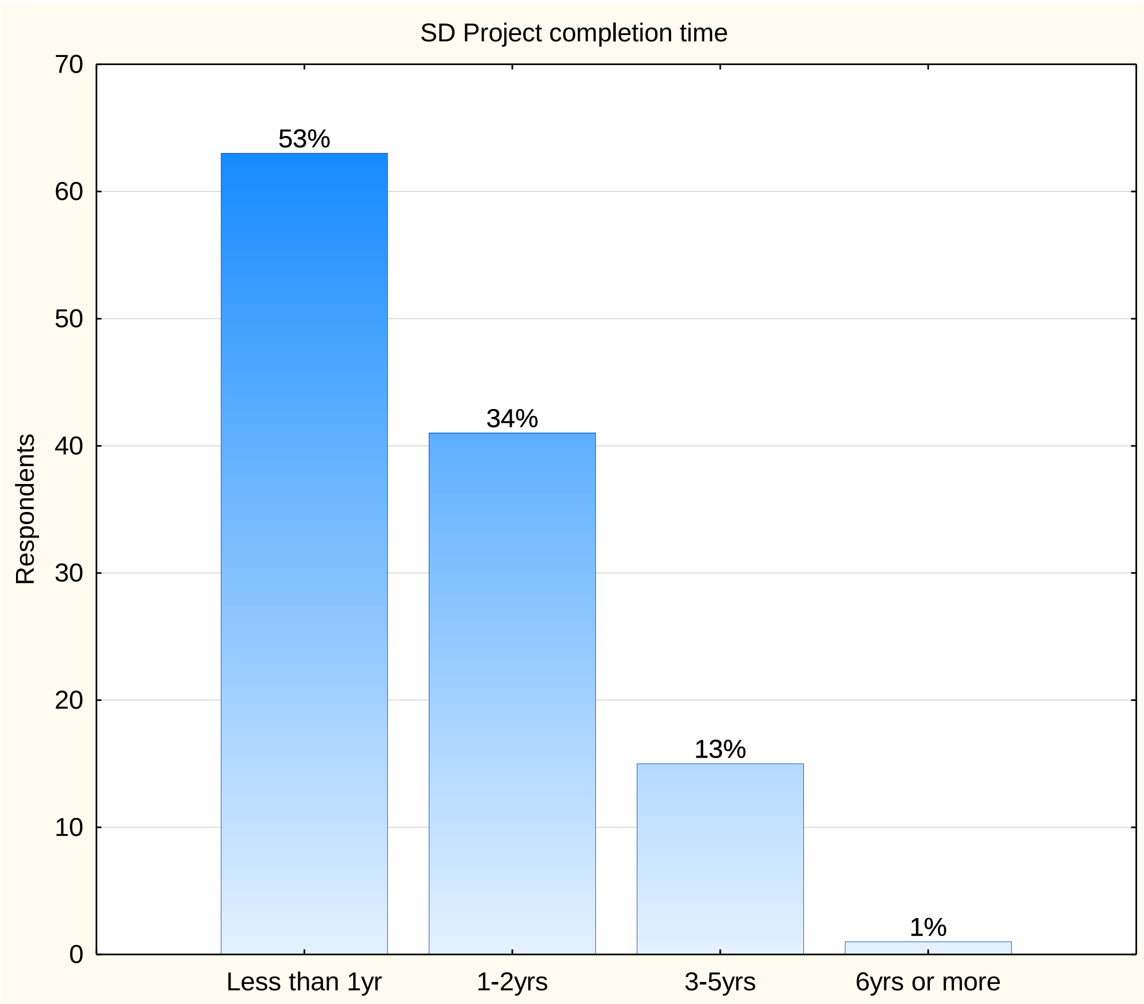


Figure 4.10: Completion time

4.6.3 The project outcome

In this question extorted from the questionnaire, it was asked that the respondents should describe the outcome of the last projects they had been involved in. It was found that out of 113 respondents who had answered, 81% claimed that the projects were completed, implemented and still in use, 9% affirmed that the projects had been cancelled, 5% were completed but not implemented and 4% were completed, implemented but not in use. Refer to figure 4.34.

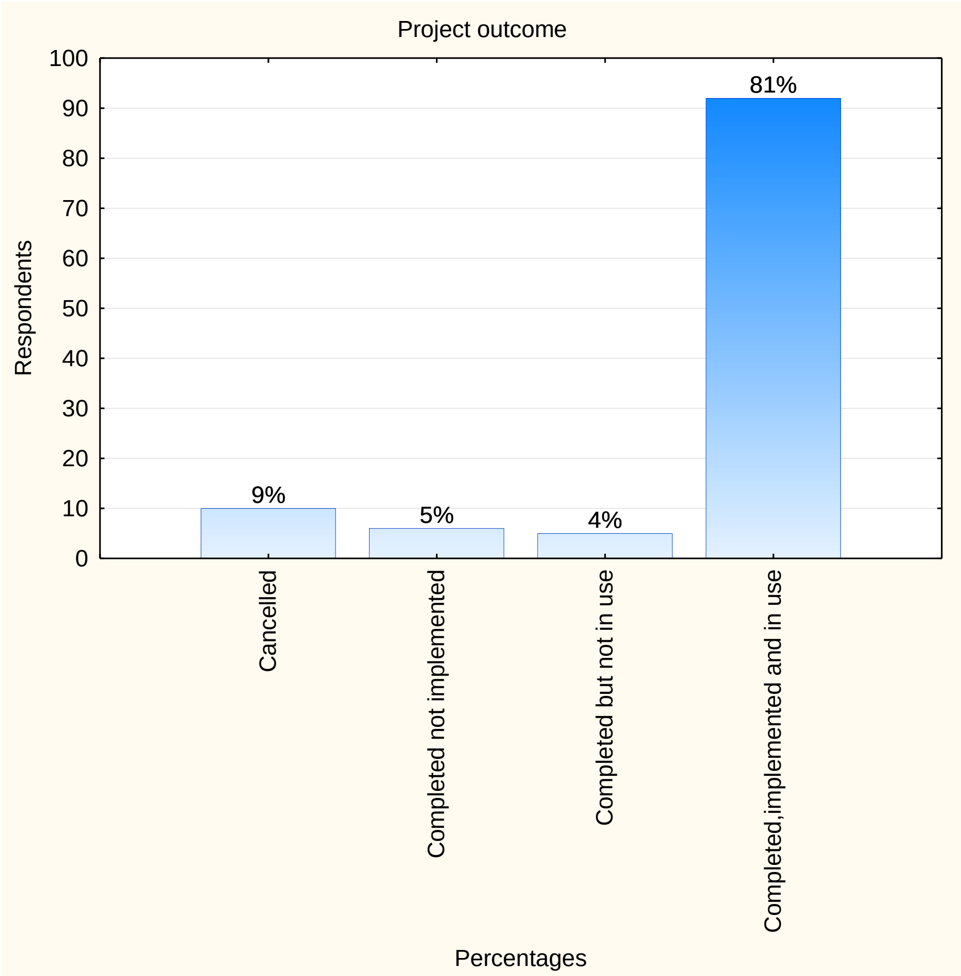


Figure 4.34: Project outcome in the last project

4.6.4 The life span of last project involved in

The question extracted from the questionnaire was phrased to determine how long the latest project the respondents had been involved in had been in use. Figure 4.35 verifies that out of 89 respondents, 15% had both two and six months using the latest project. Subsequently, the projects lasted 24 months, which involves 10% of respondents, 12 months with 9%. 1% had indicated a ‘few weeks’. The longest period was 120 months (1%).

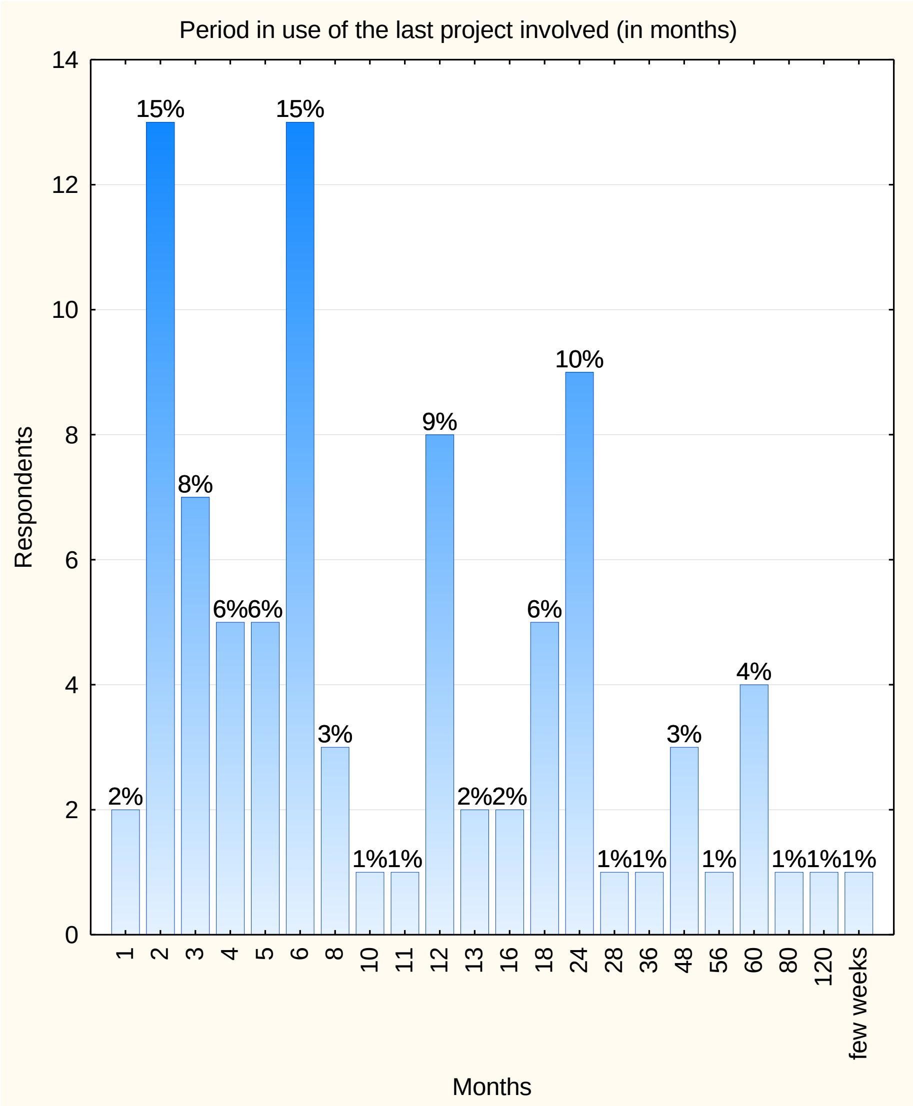


Figure 4.35: Life-span of SDMs

4.6.5 The success of the project: processes followed in the development of the project

The question from the questionnaire was posed to determine the successful processes followed in the last projects involved. Overall, Figure 4.36 shows that most of respondents agreed that the underlying factors in Table 4.6a contributed to success of project development. Table 4.36a displays the contributing factors arranged in order of the lowest mean. The most favourable factor was that "the projects were successful" (this factor obtained a mean of 1.96). The least favourable factor, suggesting that "the projects had been completed on schedule" rated a lower mean of 2.79.

Process factor that influences the project success	Totally Agree	Agree	Neutral	Disagree	Totally disagree	Mean	Std Dev
The project was successful.	30%	50%	14%	4%	1%	1.96	0.84
The project achieved its goal.	26%	59%	9%	7%	0%	1.97	0.79
The project represents excellent work.	20%	52%	20%	6%	2%	2.17	0.89
The developed system satisfied all the stated requirements.	16%	57%	11%	17%	0%	2.28	0.92
The productivity involved in the projects was high.	13%	58%	19%	8%	3%	2.30	0.90
The cost of project is low when compared to the size and complexity of the system developed.	17%	47%	21%	12%	9%	2.40	1.02
Speed of developing new applications was high.	11%	47%	21%	17%	5%	2.58	1.05
The project was completed within the budget.	9%	42%	23%	20%	6%	2.73	1.07
The project was completed on schedule.	9%	44%	14%	28%	6%	2.79	1.13

Table 4.36a: Process factors influenced the project success % and mean

4.6.5.1 Factor analysis and reliability

Table 4.36b represents the un-rotated factor loadings to determine the nature of the construct influencing observed process success variables. From Table 4.36b, the outcome of the project was measured. F1 is made up from the items that reflect the time and cost management process, so it is named “process cost and time management”. F2 is made up of the items that illustrate the goal achievement process, so it is named “process objectives and goals”. Furthermore, the Cronbach alpha was measured to determine the reliability. The reliabilities of factors are shown to be good (both 0.85), thus proving that the validity of the items is good.

Factor & Cronbach alpha	Factor Name	Process factors	Variables	Factor loading
F1- 0.85	Process cost and time management	The project was completed within the budget	E3.2	0.94
		Speed of developing new applications was high.	E3.4	0.81
		The project was completed on schedule	E3.1	0.78
		The productivity involved in the projects was high	E3.5	0.71

		The cost of the project is low when compared to the size and complexity of the system developed	E3.6	0.66
F2- 0.85	Process objectives and goals	The project achieved its goal	E3.7	0.91
		Overall, the project was successful	E3.9	0.88
		Overall, the project represents excellent work	E3.8	0.81
		The developed system satisfied all the stated requirements	E3.3	0.54

Table 4.36b: Project success process analysis and reliability

4.6.6 The success of the project: system

One of questions from the questionnaire was intended to assess the success of the completed system of the last projects involved. Table 4.37a reveals that most respondents agreed that the underlying factors were significant. The factors were arranged in terms of their ranking of means to signify their importance. The most popular factor (which obtained a mean of 1.91) indicated that the functionality of the development system was high, whereas, the highest mean of 2.83 was assigned to the documentation of the project, which was good.

Project outcome Factors	Totally Agree	Agree	Neutral	Disagree	Totally Agree	Mean	Std. Dev
The functionality of the developed system is high.	28%	58%	11%	3%	1%	1.91	0.76
The usability of the developed system is high.	26%	55%	15%	2%	2%	1.98	0.82
The developed system meets users' needs	23%	62%	9%	4%	2%	1.99	0.81
The developed system is a success.	24%	58%	13%	3%	2%	2.00	0.82
The reliability of the developed system is high.	25%	55%	17%	3%	1%	2.10	0.79
The quality of the developed system is high.	16%	68%	13%	1%	3%	2.08	0.77
The efficiency of the developed system is high.	20%	57%	15%	7%	1%	2.11	0.84
The maintainability of the developed system is high.	21%	52%	22%	4%	1%	2.12	0.82
The users are satisfied with the developed system.	17%	60%	17%	5%	2%	2.16	0.83
The portability of the developed system is high.	19%	41%	23%	10%	7%	2.44	1.12
The documentation of the developed system is good	10%	38%	21%	22%	9%	2.83	1.15

Table 4.37a: Project outcome functionality's means

4.6.6.1 Factor analysis and reliability

Table 4.37b represents the un-rotated factor loadings to determine the nature of the construct influencing observed projects success variable. From Table 4.37b, the outcome of the project was measured. F1 is deduced from the items that resemble the efficiency of the system and indicates that the project achieved its goal, thus, it is named “system success”. F2 is deduced from the items that exemplify the documentation, maintainability and portability of the system, therefore it is named “system documentation, portability and maintainability”. Furthermore, the Cronbach alpha was measured to determine the reliability. The reliabilities of factors are shown to be high (0.93 and 0.74), thus prove that the validity of the items is good.

Factor & Cronbach alpha	Factor Name	Factors influencing the product	Variables	Factor loading
F1 – 0.93	System success	The developed system meets users’ needs	E4.7	0.95
		The usability of the developed system is high	E4.6	0.89
		Overall, the users are satisfied with the developed system.	E4.10	0.87
		Overall, the developed system is a success.	E4.11	0.87
		Overall, the quality of the developed system is high	E4.9	0.80
		The reliability of the developed system is high	E4.2	0.70
		The functionality of the developed system is high	E4.1	0.56
F2 – 0.74	System documentation, portability and maintainability	The documentation of the developed system is good	E4.8	0.86
		The portability of the developed system is high	E4.4	0.73
		The maintainability of the developed system is high	E4.3	0.65
		The efficiency of the developed system is high	E4.5	0.45

Table 4.37b: Project outcome functionality analysis and reliability

4.6.6.2 Statistical analysis of factor analysis components for project outcome

In Table 4.38, the components of project success intended to match the significant aspects optimistically impacting on the success of the projects are shown. The factors are arranged in order of importance in terms of their means (the higher the mean, the more valuable the factor). It is evident that the “process cost project success” factor is ranked as the more favourable factor.

Success factor	Valid N	Mean	Minimum	Maximum	Std.Dev.
Process cost and time management	103	2.56	1	5	0.83
System success	104	2.38	1	5	0.74
Process goals and objectives	103	2.10	1	4.25	0.72
System documentation, portability and maintainability	105	2.02	1	5	0.67

Table 4.38: The statistics analysis of the project success factors

In summary, the main objective of this section was to quantify the outcomes of the most recent projects the respondents had been involved with. Essentially, the process factors that influence the outcome of the projects and the performance of the finished product were evaluated. Earlier, the life-span and the execution period of SDMs were also measured.

The next section is aimed at discussing the multiple regression and correlation to measure the relationship between IT systems success and critical success factors. Furthermore, the Lavene Test of Homogeneity, T-test and ANOVAs are employed to depict the strengths and also compare the correlation between different groups of critical success factors, SDMs factors and IT project success factors.

4.7 Section F - Statistical analyses aimed at measuring relationships between variables

The aim of this section is to determine the linear relationship between the dependent variables (project success factors) and the independent variables (critical success factors), furthermore, we compare and measure other biographical variables against CSFs, SDMs factors and project success factors.

We first exploit the correlation and multiple regression analyses to measure and predict the critical success factors (CSFs) that significantly contribute to the project success factors, and we analyse forward and backward stepwise regression to determine a set of independent variables that purportedly represent the best set of predictors of a particular dependent variable.

Furthermore, T-test analyses were executed to measure the variance between agile and traditional methods assembled from different types of SDMs; and difference between genders. Additionally, the Lavene Test of Homogeneity and analysis of variance were also used to quantify job category and business sector variables.

In practice, Table 4.38 shows that a project success encompasses four project success factors, namely process cost and time management, process objectives and goals, system documentation, portability and maintainability and system success. The afore-mentioned success factors were each measured against the fourteen critical success factors, viz. standard methods, project goals and objectives, user – developer collaboration, user requirements and involvement, technical adequacy and expertise, project plan and control, user training and systems deployment, project technique and developers proficiency, project complexity, complexity and risk level realization, project development and completion, conflicting requirement, project resource and top management involvement. Refer to Table 4.11a.

The following section presents the correlation matrix used to measure the relationship between CSFs and project success factors.

4.7.1 Correlation matrices between critical success factors (CSF) and project success factors

Correlations. Marked correlations are significant at p < .05000 N=75 (Case-wise deletion of missing data)							
CSFs	Systems development method impact	Process production and departmental development	Process quality and cost manage-ment	System suc-cess	System documentation portability and maintainability	Process objectives and goals	Process cost and time manage-ment
Standard methods	0.63	0.51	0.50	0.21	0.31	0.30	0.36
Project's goals and objectives	0.32	0.27	0.26	0.39	0.47	0.36	0.23
User-developer collaboration	0.45	0.45	0.50	0.40	0.57	0.54	0.49
User requirements	0.23	0.32	0.27	0.17	0.03	-0.06	0.12

and involvement							
Technical adequacy and project milestones	0.50	0.38	0.47	0.34	0.40	0.45	0.47
Project plan and control	0.57	0.59	0.51	0.32	0.44	0.48	0.46
User training and system deployment	0.33	0.30	0.40	0.25	0.37	0.30	0.30
Developer's expertise	0.29	0.29	0.41	-0.06	0.00	0.03	0.34
Project complexity	-0.10	-0.06	-0.10	-0.10	0.07	0.12	0.26
Project complexity & risk realization	0.25	0.33	0.27	0.19	0.21	0.23	0.48
Project development and completion	0.58	0.46	0.44	0.35	0.33	0.32	0.34
Conflicting requirement	-0.02	0.05	0.05	0.13	-0.06	0.01	0.04
Project resources	0.26	0.18	0.09	0.44	0.22	0.19	0.29
Top management Involvement	0.43	0.28	0.27	0.31	0.33	0.29	0.18

Table 4.39: Correlation between CSFs and Project success

In Table 4.39, project success factors such as "process cost and time management" show a significant correlation with all critical success factors except such factors like user requirements and involvement, conflicting requirements and top management involvement. Both project success factors such as process requirements and goals and system documentation, portability and maintainability signify the significant correlation with all critical success factors apart from user requirements and involvement, project technique and developer's proficiency, project complexity, conflicting requirements and project resources. Alternatively, project success factors such as system documentation, portability and maintainability did not designate correlation with the critical success factor vs. complexity and risks realization. Finally, project success factors such as "system success" did not denote a significant correlation with the critical success factor such as standard method, user requirements and involvement, project technique and developer's proficiency, project complexity and conflicting requirements.

The subsequent section presents the multiple regressions to determine the relationship between CSFs and IT project success factors.

4.7.2 Multiple regression between project success factors and CSFs

This section describes the multiple regressions to measure the extent to which CSFs influence project success factors. Only the p-value for the F-test of the critical success factor will be statistically significant (as $p < .05$), which indicated the significant prediction of project success factor. Each project success factor is measured against the CSFs which have shown a significant correlation with them as per correlation analysis in Table 4.39.

4.7.2.1 "Process cost and time management success factor" against CSFs

In Table 4.40, all factors are included simultaneously, and it was found that only the p-value for the F-test of the critical success factor such as "project complexity and risks realization" was significant (as $p < .05$), which signifies a unique contribution in addition to other factors; thus indicating that it significantly predicts the process cost and time management project success factor.

Forward and backward stepwise regression were employed; we selected all the factors with low p-values (less than two) for the F-tests in an attempt to develop the best model. Finally, with regard to forward stepwise regression in Table 4.40a, it was observed that three critical success factors namely technical adequacy and project milestones, project complexity and risks realization and user training and system deployment were statistically significant with R square change of 26%, 5% and 4% respectively.

Inversely, backward stepwise was exploit and critical success factors like "project plan and control" came out to be significant with the utilization of factors including project "complexity and risks realization". Eventually, when the factors were added consecutively, "user training and system deployment" was proved to be a good predictor of project success factor like "process cost and time management". Refer to Table 4.40c.

In comparison, the adjusted R square of the multiple regression models for forward and stepwise indicated the predictors and explained 32% and 31% variances respectively.

Regression Summary for Dependent Variable: process cost and time management R= .63088249 R^2= .39801272 Adjusted R^2= .32101435 F(11,86)=5.1691 p<.00000 Std.Error of estimate: .69226						
N = 98	b*	Std.Err. - of b*	b	Std.Err. - of b	t(86)	p-value
Intercept			0.41	0.37	1.08	0.28
Standard methods	-0.14	0.14	-0.14	0.15	-0.96	0.34
Project's goals and objectives	-0.16	0.12	-0.21	0.16	-1.34	0.18
User-developer collaboration	-0.09	0.14	-0.12	0.19	-0.63	0.53
User requirements and involvement	0.21	0.15	0.37	0.26	1.42	0.16
Technical adequacy and project milestones	0.24	0.14	0.31	0.18	1.73	0.09
Project plan and control	0.17	0.11	0.25	0.17	1.52	0.13
Developer's expertise	-0.01	0.11	-0.01	0.16	-0.06	0.95
Project complexity	0.03	0.09	0.03	0.10	0.35	0.73
Project complexity & risk realization	0.24	0.11	0.27	0.12	2.18	0.03
Project development and completion	0.23	0.16	0.29	0.19	1.51	0.14
Project resources	0.00	0.10	0.00	0.11	0.00	1.00

Table 4.40: Multiple regression – Process cost and time management factor against CSFs

Summary of forward Stepwise Regression; DV: process cost and time management							
Variable	Step - +in/-out	Multiple - R	Multiple - R-square	R-square - change	F - to - entr/rem	p-value	Variables - included
Technical adequacy and project milestones	1	0.51	0.26	0.26	32.87	0.00	1
Project complexity & risk realization	2	0.56	0.31	0.05	7.46	0.01	2
User training and system deployment	3	0.59	0.35	0.04	6.23	0.01	3
Project plan and control	4	0.61	0.37	0.02	2.28	0.13	4
Project's goals and objectives	5	0.61	0.38	0.01	1.47	0.23	5
Project development and completion	6	0.62	0.39	0.01	1.53	0.22	6

Table 4.40a: Forward stepwise regression – Process cost and time management factor against CSFs

Regression Summary for Dependent Variable: process cost and time management R= .62284240 R^2= .38793266 Adjusted R^2= .34757657 F(6,91)=9.6127 p						
	b*	**Std.Err. - of b***	**b**	**Std.Err. - of b**	**t(91)**	**p-value**
Intercept			0.49	0.32	1.52	0.13
Technical adequacy and project milestones	0.19	0.14	0.33	0.24	1.36	0.18
Project complexity and risk realization	0.21	0.10	0.23	0.10	2.16	0.03
User training and system deployment	0.17	0.11	0.26	0.16	1.65	0.10
Project plan and control	0.17	0.12	0.22	0.16	1.40	0.16
Project's goals and objectives	-0.17	0.11	-0.23	0.15	-1.55	0.12
Project development and completion	0.16	0.13	0.20	0.16	1.24	0.22

Table 4.40b: Multiple regression (forward) – Process cost and time management factor against CSFs

Summary of backward Stepwise Regression; DV: process cost and time management							
Variable	Step - +in/-out	Multiple - R	Multiple - R-square	R-square - change	F - to - entr/rem	p-value	Variables - included
Project resources	-1	0.63	0.40	0.00	0.00	1.00	10
Developer's expertise	-2	0.63	0.40	0.00	0.00	0.95	9
Project complexity	-3	0.63	0.40	0.00	0.13	0.72	8
User-developer collaboration	-4	0.63	0.39	0.00	0.39	0.53	7
Standard methods	-5	0.62	0.39	-0.01	0.97	0.33	6
Project development and completion	-6	0.61	0.38	-0.01	1.53	0.22	5
Project's goals and objectives	-7	0.61	0.37	-0.01	1.47	0.23	4
Technical adequacy and project milestones	-8	0.60	0.36	-0.01	1.73	0.19	3
User training and system deployment	-9	0.57	0.32	-0.03	5.08	0.03	2

Table 4.40c: Backward stepwise regression – Process cost and time management factor against CSFs

Regression Summary for Dependent Variable: process cost and time management R= .56669054 R^2= .32113817 Adjusted R^2= .30684634 F(2,95)=22.470 p<.00000 Std.Error of estimate: .69944						
N=98	b*	Std.Err. - of b*	b	Std.Err. - of b	t(95)	p-value
Intercept			0.82	0.27	3.03	0
Project plan and control	0.38	0.09	0.48	0.11	4.21	0
Project complexity & risk realization	0.3	0.09	0.33	0.1	3.36	0

Table 4.40d: Multiple regression (backward) – Process cost and time management factor against CSFs

4.7.2.2 “Process goals and objectives” success factor against CSFs

In Table 4.41, all factors are included simultaneously, but only the p-value for the F-test of critical success factor such as “project plan and control” was significant (as $p < .05$) which signifies a unique contribution in addition to other factors; thus indicating that it significantly predicts the “process goals and objectives” for a project’s success factor.

Since the analysis did not indicate a feasible model, forward and backward stepwise regressions were employed; we selected all the factors with low p-values (less than two) for the F-test in an attempt to develop the best model. Finally, with regard to forward stepwise regression in Table 4.41a, it was observed that critical success factors such as project plan and control and user-developer collaboration were statistically significant to process goals and objectives success factor, therefore explicating the R square change of 25% and 3% respectively.

Furthermore, with the deployment of the backward stepwise regression, the result still shows that a critical success factor like project plan and control was a good predictor of process goals and objectives. Alternatively, with the consecutive application of the factors, user-developer collaboration was statistically significant to process goals and objectives success factor (refer to Table 4.41c).

In comparison, the adjusted R square of the multiple regression models for forward and backward stepwise indicated that the predictors explained 26% and 25% variances respectively.

Regression Summary for Dependent Variable: process goals and objectives R= .57598209 R^2= .33175537 Adjusted R^2= .25832189 $F(10,91)$=4.5178 $p<.00003$ Std.Error of estimate: .61975						
N-102	b*	Std.Err. - of b*	b	Std.Err. - of b	t(91)	p-value
Intercept			0.54	0.35	1.54	0.13
Standard methods	-0.20	0.14	-0.18	0.12	-1.44	0.15
Project’s goals and objectives	0.07	0.12	0.08	0.14	0.57	0.57
User-developer collaboration	0.18	0.14	0.21	0.17	1.27	0.21

Technical adequacy and project milestones	-0.10	0.16	-0.15	0.23	-0.64	0.53
Project plan and control	0.37	0.14	0.39	0.15	2.62	0.01
User training and system deployment	0.08	0.11	0.11	0.14	0.76	0.45
Project complexity & risk realization	0.08	0.11	0.07	0.10	0.67	0.51
Project development and completion	0.05	0.15	0.06	0.16	0.35	0.73
Conflicting requirement	-0.02	0.09	-0.02	0.07	-0.24	0.81
Top management Involvement	0.17	0.13	0.19	0.14	1.29	0.20

Table 4.41: Multiple regression – Process goal and objective factor against CSFs

Summary of forward stepwise regression; DV: process goals and objectives							
Variable	Step - +in/- out	Multiple - R	Multiple - R- square	R-square - change	F - to - entr/rem	p- value	Variables - included
Project plan and control	1	0.50	0.25	0.25	34.19	0.00	1
User-developer collaboration	2	0.54	0.29	0.03	4.39	0.04	2
Top management Involvement	3	0.55	0.30	0.02	2.27	0.14	3
Standard methods	4	0.56	0.32	0.01	1.86	0.18	4
User training and system deployment	5	0.57	0.32	0.01	1.15	0.29	5

Table 4.41a: Forward stepwise regression – Process goal and objective factor against CSFs

Regression summary for dependent variable: process goals and objectives R= .56903833 R²= .32380462 Adjusted R²= .28858611 F(5,96)=9.1942 p						
	b*	**Std.Err. - of b***	**B**	**Std.Err. - of b**	**t(96)**	**p-value**
Intercept			0.49	0.26	1.92	0.06
Project plan and control	0.35	0.13	0.36	0.14	2.69	0.01
User-developer collaboration	0.20	0.12	0.23	0.14	1.71	0.09
Top management Involvement	0.18	0.11	0.20	0.12	1.64	0.10
Standard methods and	-0.18	0.12	-0.16	0.11	-1.52	0.13
User training and system deployment	0.11	0.10	0.14	0.13	1.07	0.29

Table 4.41b: Multiple regression (forward) – Process goal and objective factor against CSFs

Summary of backward stepwise regression; DV: Process goal and objective							
Variable	Step - +in/-out	Multiple – R	Multiple - R-square	R-square - change	F - to - entr/rem	p-value	Variables - included
Conflicting requirement	-1	0.58	0.33	0.00	0.06	0.81	9
Project development and completion	-2	0.57	0.33	0.00	0.11	0.74	8
Technical adequacy and project milestones	-3	0.57	0.33	0.00	0.37	0.54	7
Project's goals and objectives	-4	0.57	0.33	0.00	0.37	0.55	6
Project complexity & risk realization	-5	0.57	0.32	0.00	0.20	0.65	5
User training and system deployment	-6	0.56	0.32	-0.01	1.15	0.29	4
Standard methods	-7	0.55	0.30	-0.01	1.86	0.18	3
Top management Involvement	-8	0.54	0.29	-0.02	2.27	0.14	2
User-developer collaboration	-9	0.50	0.25	-0.03	4.39	0.04	1

Table 4.41c: Backward stepwise regression – Process goal and objective factor against CSFs

Regression summary for dependent variable: process goals and objectives : R= .50477602 R^2= .25479883 Adjusted R^2= .24734682 F(1,100)=34.192 p<.00000 Std.Error of estimate: .62432						
N=102	b*	Std.Err. - of b*	b	Std.Err. - of b	t(100)	p-value
Intercept			0.88	0.22	4.03	0.00
Project plan and control	0.50	0.09	0.53	0.09	5.85	0.00

Table 4.41d: Multiple regression (backward) Process goal and objective factor against CSFs

4.7.2.3 "System documentation, portability and maintainability" success factor against CSFs

In Table 4.42, all factors are included simultaneously, only the p-value for the F-test of critical success factor such as "project user-developer collaboration" was significant (as $p < .05$) which represents a unique contribution in addition to other factors; thus indicating that it significantly predicts the “system documentation, portability and maintainability” in terms of the project success factor.

Forward and backward stepwise regression were employed; we selected all the factors with low p-values (less than two) for the F-test in an attempt to develop the best model. Finally, with regard to forward stepwise regression (refer to Table 4.42a), it was observed that critical success factors namely; “user-developer collaboration”, “project’s goals and objectives” and “conflicting requirement” were statistically significant to the system “documentation, portability and maintainability” success factor, thus embracing the R square change of 24%, 3% and 3% respectively.

With the effect of backward stepwise regression (refer to Table 4.42c), user-developer collaboration was a good predictor of system documentation, portability and maintainability. Thus, when the factors were loaded consecutively, project goals and objectives as a critical success factor was statistically significant in predicting system documentation, portability and maintainability.

In comparison, the adjusted R square of the multiple regression models for forward and stepwise regressions indicated that the predictors explained 36% and 24% variances respectively.

Regression summary for dependent variable: System documentation, portability and maintainability :R= .60072100 R^2= .36086571 Adjusted R^2= .29967201 F(9,94)=5.8971 p<.00000 Std.Error of estimate: .55806						
N--104	b*	Std.Err. - of b*	b	Std.Err. - of b	t(94)	p-value
Intercept			0.94	0.30	3.14	0.00
Standard methods	-0.17	0.13	-0.14	0.11	-1.27	0.21
Project’s goals and objectives	0.18	0.11	0.19	0.11	1.65	0.10
User-developer collaboration	0.41	0.13	0.44	0.14	3.20	0.00
Technical adequacy and project milestones	-0.25	0.15	-0.33	0.19	-1.70	0.09
Project plan and control	0.20	0.13	0.19	0.13	1.48	0.14
User training and system deployment	0.18	0.11	0.20	0.12	1.61	0.11

Project development and completion	-0.09	0.14	-0.09	0.14	-0.63	0.53
Conflicting requirement	-0.11	0.09	-0.08	0.06	-1.29	0.20
Top management Involvement	0.19	0.13	0.19	0.13	1.49	0.14

Table 4.42: Multiple regression - system documentation, portability and maintainability factor against CSFs

Summary of forward stepwise regression; DV: system documentation, portability and maintainability							
Variable	Step - +in/- out	Multiple – R	Multiple - R-square	R-square - change	F - to - entr/rem	p-value	Variables - included
User-developer collaboration	1	0.49	0.24	0.24	32.45	0.00	1
Project's goals and objectives	2	0.52	0.27	0.03	4.11	0.05	2
Conflicting requirement	3	0.55	0.30	0.03	4.18	0.04	3
User training and system deployment	4	0.56	0.31	0.01	1.78	0.19	4
Technical adequacy and project milestones	5	0.57	0.33	0.01	2.07	0.15	5
Top management Involvement	6	0.58	0.33	0.01	1.15	0.29	6
Standard methods	7	0.59	0.34	0.01	1.32	0.25	7
Project plan and control	8	0.60	0.36	0.01	2.14	0.15	8

Table 4.42a: Forward stepwise regression – system documentation, portability and maintainability against CSFs

Regression summary for dependent variable: System documentation, portability and maintainability $R= .59849732$ $R^2= .35819905$ Adjusted $R^2= .30415265$ $F(8,95)=6.6276$ p						
	b*	**Std.Err. - of b***	**b**	**Std.Err. - of b**	**t(95)**	**p-value**
Intercept			0.95	0.30	3.20	0.00
User-developer collaboration	0.42	0.13	0.44	0.13	3.28	0.00
Project's goals and objectives	0.16	0.10	0.17	0.11	1.54	0.13
Conflicting requirement	-0.12	0.09	-0.08	0.06	-1.35	0.18
User training and system deployment	0.16	0.11	0.18	0.12	1.51	0.14
Technical adequacy and project milestones	-0.27	0.14	-0.35	0.19	-1.86	0.07
Top management Involvement	0.17	0.12	0.17	0.12	1.40	0.16
Standard methods	-0.20	0.12	-0.16	0.10	-1.59	0.12
Project plan and control	0.19	0.13	0.19	0.13	1.46	0.15

Table 4.42b: Multiple regression (forward) - system documentation, portability and maintainability factor against CSFs

Summary of backward stepwise regression; DV System documentation, portability and maintainability							
Variable	Step - +in/-out	Multiple - R	Multiple - R-square	R-square - change	F - to - entr/rem	p-value	Variables - included
Project development and completion	-1	0.60	0.36	0.00	0.39	0.53	8
Conflicting requirement	-2	0.59	0.35	-0.01	1.83	0.18	7
Top management Involvement	-3	0.58	0.33	-0.01	2.12	0.15	6
Technical adequacy and project milestones	-4	0.56	0.32	-0.02	2.28	0.13	5
Project plan and control	-5	0.55	0.31	-0.01	1.50	0.22	4
Standard methods	-6	0.54	0.29	-0.02	2.16	0.15	3
User training and system deployment	-7	0.52	0.27	-0.02	2.68	0.11	2
Project's goals and objectives	-8	0.49	0.24	-0.03	4.11	0.05	1

Table 4.42c: Backward stepwise regression – system documentation, portability and maintainability against CSFs

Regression summary for dependent variable: System documentation, portability and maintainability: R= .49129076 R²= .24136661 Adjusted R²= .23392903 F(1,102)=32.452 p<.00000 Std.Error of estimate: .58367						
N-104	b*	Std.Err. - of b*	b	Std.Err. - of b	t(102)	p-value
Intercept			1.01	0.19	5.36	0.00
User-developer collaboration	0.49	0.09	0.52	0.09	5.70	0.00

Table 4.42d: Multiple regression (backward) - system documentation, portability and maintainability against CSFs

4.7.2.4 "System success factors" against CSFs

In Table 4.43, all factors are included simultaneously, but only the p-value for the F-test of critical success factor such as "project resources" was significant (as $p < .05$) which demonstrates a unique contribution in addition to other CSFs; thus indicating that it is a significant predictor of the system success factor.

Forward and backward stepwise regressions were employed; we selected all the factors with low p – values (less than two) for the t-tests in an attempt to develop the best model. Typically, with the application of forward stepwise regression (refer to Table 4.43a), it was observed that critical success factors like "project resources" and "user training and system deployment" were statistically significant to the system success factor, thus obtaining the R square change of 16% and 7% respectively.

With the exercise of backward stepwise regression (refer to Table 4.43c and Table 4.43d), the critical success factors such as "project resources" and "project plan and control" were statistically significant in that they predicted the system success factor.

In comparison, the adjusted R square of the multiple regression models for forward and two backward stepwise models indicated that the predictors explained 21% and 23 and 15% variances respectively.

Regression summary for dependent variable: System success: R= .53499101 R^2= .28621538 Adjusted R^2= .20863010 F(10,92)=3.6890 p<.00035 Std. Error of estimate: .66270						
N-103	b*	Std. Err. - of b*	b	Std. Err. - of b	t(92)	p-value
Intercept			0.86	0.37	2.32	0.02
Standard methods	-0.18	0.14	-0.17	0.13	-1.30	0.20
Project's goals and objectives	-0.07	0.12	-0.09	0.14	-0.62	0.54
User-developer collaboration	0.14	0.14	0.17	0.16	1.04	0.30
Technical adequacy and project milestones	-0.19	0.16	-0.29	0.24	-1.23	0.22
Project plan and control	0.20	0.14	0.22	0.16	1.41	0.16
User training and system deployment	0.18	0.12	0.23	0.15	1.56	0.12
Project development and completion	0.20	0.15	0.22	0.17	1.31	0.19
Conflicting requirement	0.02	0.09	0.02	0.07	0.22	0.83
Top management Involvement	0.14	0.13	0.16	0.15	1.08	0.28
Project resources	0.23	0.10	0.23	0.10	2.21	0.03

Table 4.43: Multiple regression – system success factor against CSFs

Summary of forward stepwise regression; DV: System success							
Variable	Step - +in/- out	Multiple - R	Multiple - R-square	R-square - change	F - to - entr/rem	p-value	Variables - included
User training and system deployment	1	0.40	0.16	0.16	18.98	0.00	1
Project resources	2	0.47	0.23	0.07	8.66	0.00	2
Project plan and control	3	0.50	0.25	0.02	3.02	0.09	3

Table 4.43a: Forward stepwise regression – system success against CSFs

Regression summary for dependent variable: System success: R= .49819695 R^2= .24820021 Adjusted R^2= .22541839 F(3,99)=10.895 p<.00000 Std. Error of estimate: .65564						
N-103	b*	Std. Err. - of b*	b	Std. Err. - of b	t(99)	p-value
Intercept			0.86	0.28	3.11	0.00
User training and system deployment	0.21	0.11	0.27	0.13	2.03	0.05
Project resources	0.25	0.09	0.25	0.10	2.67	0.01
Project plan and control	0.18	0.10	0.20	0.11	1.74	0.09

Table 4.43b: Multiple regression (forward) - system success against CSFs

Summary of backward stepwise regression; DV: System success							
Variable	Step - +in/- out	Multiple – R	Multiple - R-square	R-square - change	F - to - entr/rem	p-value	Variables - included
Conflicting requirement	-1	0.53	0.29	0.00	0.05	0.83	9
Project's goals and objectives	-2	0.53	0.28	0.00	0.38	0.54	8
User-developer collaboration	-3	0.53	0.28	-0.01	0.91	0.34	7
Technical adequacy and project milestones	-4	0.52	0.27	-0.01	1.04	0.31	6
Top management Involvement	-5	0.51	0.26	0.00	0.60	0.44	5

Standard methods	-6	0.51	0.26	-0.01	1.10	0.30	4
Project development and completion	-7	0.50	0.25	-0.01	0.92	0.34	3
Project plan and control	-8	0.47	0.23	-0.02	3.02	0.09	2
Project resources	-9	0.40	0.16	-0.07	8.66	0.00	1

Table 4.43c: Backward stepwise regression – system success against CSFs

Regression summary for dependent variable: System success: $R= .39770504$ $R^2= .15816930$ Adjusted $R^2= .14983434$ $F(1,101)=18.977$ $p<.00003$ Std .Error of estimate: .68688						
N-103	b*	Std. Err. - of b*	b	Std. Err. - of b	t(101)	p-value
Intercept			1.40	0.24	5.93	0.00
User training and system deployment	0.40	0.09	0.50	0.11	4.36	0.00

Table 4.43d: Multiple regression (backward) - system success against CSFs

4.7.3 T-test to measure agile and traditional SDMs and genders

4.7.3.1 Agile and traditional SDMs

The extent to which SDMs were used in terms of benefits and not being used was measured in Section C and Section D of chapter4. These SDMs were identified and grouped between agile and traditional methods. Thus, the aim of this sub-section is to examine the difference of means between agile and traditional methods contributing towards CSFs, SDMs factors (quality and process), project success factors (process and system) and SDMs non-use reason factors.

A T-test was employed and the p-value was set to be statistically significant at p-value for F-test ($p < 0.5$). From the table 4.44, it is evident that there is a significant difference of mean between traditional and agile methods on the factors namely; “project technique and developer’s proficiency”, “level of complexity and risks realization”, SDMs quality cost and project success execution. Typically, the results suggested that “project technique and developer’s proficiency”, “level of complexity and risks realization”, “quality cost and system execution” success factor were carried out through the usage of the traditional and agile method hence their means were proven statistically difference.

T-tests; Grouping: Other SDMs_Recode Group 1: agile Group 2: Traditional											
Variable	Mean agile	Mean – Trad-itional	t-value	Df	p	Valid N - agile	Valid N – Tradi-tional	Std. Dev - agile	Std. Dev. – Tradi-tional	F-ratio – Vari-ances	p – Vari-ances
Standard methods	2.11	1.75	1.56	36.00	0.13	25.00	13.00	0.68	0.65	1.10	0.90
Project's goals and objectives	1.91	1.92	-0.06	36.00	0.95	25.00	13.00	0.63	0.66	1.09	0.82
User-developer collaboration	1.90	1.74	1.01	36.00	0.32	25.00	13.00	0.43	0.57	1.76	0.23
User requirements and involvement	2.22	1.88	1.80	36.00	0.08	25.00	13.00	0.45	0.70	2.49	0.06
Technical adequacy and project milestones	1.94	1.75	1.14	36.00	0.26	25.00	13.00	0.39	0.64	2.66	0.04
Project plan and control	2.10	2.12	-0.07	36.00	0.95	25.00	13.00	0.65	0.67	1.07	0.85
User training and system deployment	2.09	1.83	1.41	36.00	0.17	25.00	13.00	0.54	0.57	1.11	0.79
Developer's expertise	1.97	1.64	2.42	36.00	0.02	25.00	13.00	0.39	0.42	1.18	0.70
Project complexity	2.17	1.75	1.42	34.00	0.17	24.00	12.00	0.87	0.75	1.33	0.64
Project complexity & risk realization	2.04	1.54	2.19	36.00	0.04	25.00	13.00	0.73	0.52	2.01	0.21
Project development and completion	2.21	1.83	1.96	36.00	0.06	25.00	13.00	0.54	0.64	1.38	0.48
Conflicting requirements	2.32	2.23	0.30	36.00	0.77	25.00	13.00	0.90	0.83	1.17	0.80
Project resources	2.26	2.04	0.93	36.00	0.36	25.00	13.00	0.72	0.63	1.33	0.62
Top management Involvement	1.97	1.79	0.92	36.00	0.36	25.00	13.00	0.57	0.63	1.23	0.64
Systems development method impact	2.19	2.06	0.63	34.00	0.53	24.00	12.00	0.53	0.68	1.67	0.29

Process production and departmental development	2.19	2.05	0.81	34.00	0.42	24.00	12.00	0.50	0.47	1.16	0.83
Process quality improvement and cost reduction	2.56	2.08	2.30	34.00	0.03	24.00	12.00	0.62	0.49	1.56	0.44
Resource deficiency and benefit uncertainty	3.09	3.14	-0.14	12.00	0.89	5.00	9.00	0.56	0.80	2.06	0.51
Existence method competency and cost accumulation	2.80	2.22	1.55	12.00	0.15	5.00	9.00	0.76	0.62	1.51	0.58
Ambiguity and irrelevancy of SDMs	3.10	2.97	0.29	12.00	0.78	5.00	9.00	0.84	0.78	1.18	0.78
Process cost and time management	2.69	2.28	1.26	28.00	0.22	20.00	10.00	0.96	0.46	4.27	0.03
Process objectives and goals	2.15	2.18	-0.10	28.00	0.92	20.00	10.00	0.67	0.55	1.45	0.58
System documentation, portability and maintainability	2.11	2.00	0.57	28.00	0.57	20.00	10.00	0.51	0.48	1.12	0.90
System success	2.61	2.23	2.06	28.00	0.05	20.00	10.00	0.50	0.45	1.26	0.75

Table 4.44: T-Tests: Agile and traditional SDMs

4.7.3.2 The degree of adoption of SDMs

The degree to which SDMs were adopted was measured in Section C and Section D of chapter4 (the benefit and reason for not using SDMs). These SDMs were identified and grouped between agile and traditional methods. Thus, the aim of this sub-section is to examine the difference of means between agile and traditional methods with regard to their adoption in relation to CSFs, SDMs factors (quality and process), project success factors (product and system) and SDMs non-use reason.

A T-test was employed and a p-value was set to be statistically significant as p-value for F-test ($p < 0.5$). From the table4.45, it is evident that there is a significant difference of means between traditional and agile methods on the quality cost factor of SDMs. Typically, the results suggest that “process quality improvement and cost reduction” factor was a strong contribution to the adoption made in the process to traditional and agile methods hence it assumed to be statistical differences of means between agile and traditional methods.

T-tests; Grouping: Other SDMs SPECIFIED_RECODING Group 1: Agile Group 2: Traditional											
Variable	Mean - agile	Mean – Tradi-tional	t-value	df	p	Valid N - agile	Valid N – Tradi-tional	Std. D - agile	Std. D – Tradi-tional	F-ratio – Vari-ances	p – Vari-ances
Standard methods	2.00	1.88	0.35	20.00	0.73	18.00	4.00	0.63	0.75	1.42	0.54
Project's goals and objectives	2.06	2.31	-0.65	20.00	0.52	18.00	4.00	0.66	0.94	2.03	0.29
User-developer collaboration	1.98	2.20	-0.67	20.00	0.51	18.00	4.00	0.61	0.54	1.25	0.98
User requirements and involvement	2.26	2.19	0.27	20.00	0.79	18.00	4.00	0.47	0.66	1.93	0.33
Technical adequacy and project milestones	1.95	2.13	-0.59	20.00	0.56	18.00	4.00	0.49	0.75	2.34	0.22
Project plan and control	2.19	2.44	-0.57	20.00	0.57	18.00	4.00	0.70	1.05	2.22	0.25
User training and system deployment	2.17	1.94	0.81	20.00	0.43	18.00	4.00	0.48	0.72	2.23	0.24
Developer's expertise	2.00	1.83	0.69	20.00	0.50	18.00	4.00	0.42	0.53	1.57	0.46
Project complexity	2.06	1.75	0.78	19.00	0.45	17.00	4.00	0.75	0.50	2.24	0.55
Project complexity & risk realization	2.00	2.25	-0.53	20.00	0.60	18.00	4.00	0.77	1.26	2.69	0.16
Project development and completion	2.21	2.05	0.50	20.00	0.62	18.00	4.00	0.52	0.81	2.36	0.22
Conflicting requirement	2.39	2.25	0.27	20.00	0.79	18.00	4.00	0.92	0.96	1.09	0.76
Project resources	2.06	2.75	-1.85	20.00	0.08	18.00	4.00	0.64	0.87	1.84	0.36
Top management	1.99	1.81	0.49	20.00	0.63	18.00	4.00	0.59	0.90	2.32	0.22

Involvement											
Systems development method impact	2.19	1.69	1.83	20.00	0.08	18.00	4.00	0.43	0.77	3.19	0.10
Process production and departmental development	2.29	2.07	0.82	20.00	0.42	18.00	4.00	0.42	0.78	3.39	0.08
Process quality improvement and cost reduction	2.56	1.75	2.64	20.00	0.02	18.00	4.00	0.55	0.57	1.08	0.77
Resource deficiency and benefit uncertainty	3.20	2.29	2.30	5.00	0.07	5.00	2.00	0.52	0.20	6.65	0.56
Existence method competency and cost accumulation	2.60	2.25	0.61	5.00	0.57	5.00	2.00	0.74	0.35	4.40	0.68
Ambiguity and irrelevancy of SDMs	3.55	3.00	0.74	5.00	0.49	5.00	2.00	0.84	1.06	1.61	0.55
Process cost and time management	2.76	2.80	-0.06	15.00	0.95	15.00	2.00	0.83	1.41	2.89	0.22
Process objectives and goals	2.23	1.75	0.88	15.00	0.39	15.00	2.00	0.75	0.35	4.46	0.71
System documentation, portability and maintainability	2.36	2.00	0.69	15.00	0.50	15.00	2.00	0.72	0.20	12.70	0.43
System success	2.53	3.00	-1.04	15.00	0.32	15.00	2.00	0.55	1.06	3.72	0.15

Table 4.45: T-Tests: Degree of adoption of SDMs

4.7.3.3 The response between genders

The responses in terms of genders were observed in Section B (biographical), and attention was given to the way in which males and females were depicted vis-à-vis underlying critical success and project success factors. Particularly, the aim of this sub-section is to examine the difference of means between genders towards CSFs, SDM factors (quality and process), project success factors (product and system) and SDMs non-use reason factors.

A T-test was employed and a p-value set to be statistically significant at p-value for F-test ($p < 0.5$). From the table 4.46, it is evident that there is a significant difference of means between the males and females on factors that gain the p-value for the t-test ($p < 0.05$). Sensibly, the results suggested that the means of females and males were statistically different to the following factors namely "user-developer collaboration", "user requirements and involvement", "top management involvement", "resource deficiency and benefit uncertainty" and "ambiguity and irrelevancy of SDMs".

T-tests; Grouping: Gender Group 1: Male Group 2:Female										
Variable	Mean - Male	Mean – Fe-male	t-value	df	p	Valid N - 1	Valid N - 2	Std._ Dev. - 1	Std._ Dev. - 2	F-ratio – Va-riance	p – Va-riance
Standard methods	2.14	1.75	1.91	115.00	0.06	100.00	17.00	0.79	0.71	1.23	0.67
Project's goals and objectives	1.92	1.82	0.55	114.00	0.59	99.00	17.00	0.64	0.69	1.17	0.60
User-developer collaboration	2.07	1.66	2.51	114.00	0.01	99.00	17.00	0.64	0.48	1.76	0.20
User requirements and involvement	2.37	2.00	2.20	114.00	0.03	99.00	17.00	0.65	0.60	1.17	0.75
Technical adequacy and project milestones	2.01	1.81	1.49	114.00	0.14	99.00	17.00	0.51	0.62	1.47	0.25
Project plan and control	2.30	2.31	-0.03	114.00	0.97	99.00	17.00	0.67	0.75	1.25	0.49
User training and system deployment	1.99	1.90	0.61	114.00	0.54	99.00	17.00	0.60	0.53	1.30	0.57
Developer's expertise	1.87	1.81	0.36	114.00	0.72	99.00	17.00	0.62	0.65	1.09	0.74
Project complexity	2.08	1.75	1.50	110.00	0.14	96.00	16.00	0.84	0.68	1.52	0.36
Project complexity & risk realization	1.97	1.59	1.92	114.00	0.06	99.00	17.00	0.72	0.94	1.70	0.12
Project development and completion	2.15	1.89	1.44	115.00	0.15	100.00	17.00	0.69	0.62	1.25	0.64

Project development and completion	2.49	2.35	0.60	114.00	0.55	99.00	17.00	0.90	0.93	1.08	0.77
Project resources	2.10	2.18	-0.40	114.00	0.69	99.00	17.00	0.71	0.79	1.25	0.50
Top management Involvement	2.03	1.60	2.47	114.00	0.02	99.00	17.00	0.66	0.66	1.00	0.92
Systems development impact	2.15	2.10	0.23	108.00	0.82	93.00	17.00	0.79	0.49	2.56	0.04
Process production and departmental development	2.25	2.21	0.23	108.00	0.82	93.00	17.00	0.65	0.60	1.16	0.77
Process quality improvement and cost reduction	2.28	2.24	0.20	108.00	0.84	93.00	17.00	0.84	0.28	8.86	0.00
Resource deficiency and benefit uncertainty	2.99	3.55	-2.01	55.00	0.05	49.00	8.00	0.74	0.65	1.33	0.74
Existence method competency and cost accumulation	2.63	3.00	-1.05	55.00	0.30	49.00	8.00	0.92	0.85	1.19	0.88
Ambiguity and irrelevancy of SDMs	2.95	3.81	-2.69	55.00	0.01	49.00	8.00	0.88	0.50	3.17	0.11
Process cost and time management	2.50	2.72	-1.00	97.00	0.32	86.00	13.00	0.78	0.61	1.61	0.36
Process objectives and goals	2.12	2.02	0.44	97.00	0.66	86.00	13.00	0.75	0.59	1.61	0.36
System documentation, portability and maintainability	2.05	1.92	0.61	99.00	0.54	88.00	13.00	0.70	0.49	2.04	0.17
System success	2.35	2.35	0.03	98.00	0.97	87.00	13.00	0.73	0.76	1.09	0.76

Table 4.46: T-Tests: Response differences between genders

7.3.3.4 Lavene Test of homogeneity and ANOVA

This section provides the Lavene’s Test as used to assess the variance homogeneity of “job categories”, “business sectors” in relation to CSFs, SDMs factors (quality and process), success factors (product and system) and SDMs non-use reasons for factors. The aim is to assess the difference of variances of different groups which serve as a precondition for parametric tests such as ANOVA. The analyses will be made on variables vs. Job category and business sector. If the significance from this test is less than 0.05 (p value < 0.5), then variances are significantly different on different groups therefore parametric tests cannot be used and vice versa.

a) Lavene Test of Homogeneity of Job category

The views of project managers, developers and other professionals were examined in section A 3 of Chapter4. Thus, their variance homogeneities were tested and it was observed that job category has a significantly different variance towards factors such as "user-developer collaboration" and "system development impact", Refer to table 4.47.

Lavene Test of Homogeneity of Variances: Marked effects are significant at p < .05000 Include condition: Job category – Project Manager, Developer & other								
Variable	SS - Effect	df - Effect	MS - Effect	SS - Error	df - Error	MS – Error	**F**	**p**
Standard methods	0.11	2.00	0.05	19.95	100.00	0.20	0.27	0.76
Project's goals and objectives	0.54	2.00	0.27	14.47	99.00	0.15	1.86	0.16
User-developer collaboration	1.13	2.00	0.57	9.31	99.00	0.09	6.02	0.00
User requirements and involvement	0.48	2.00	0.24	13.51	99.00	0.14	1.75	0.18
Technical adequacy and project milestones	0.15	2.00	0.07	8.90	99.00	0.09	0.83	0.44
Project plan and control	0.58	2.00	0.29	13.42	99.00	0.14	2.16	0.12
User training and system deployment	0.43	2.00	0.22	11.01	99.00	0.11	1.95	0.15
Developer's expertise	0.51	2.00	0.25	14.35	99.00	0.14	1.75	0.18
Project complexity	0.10	2.00	0.05	21.93	95.00	0.23	0.21	0.81
Project complexity & risk realization	0.02	2.00	0.01	20.79	99.00	0.21	0.05	0.95
Project development and completion	0.09	2.00	0.04	15.60	100.00	0.16	0.28	0.75
Conflicting requirement	0.01	2.00	0.00	18.90	99.00	0.19	0.02	0.98
Project resources	0.05	2.00	0.03	16.18	99.00	0.16	0.16	0.86

Top management Involvement	0.71	2.00	0.36	14.32	99.00	0.14	2.47	0.09
Systems development method impact	3.12	2.00	1.56	21.34	93.00	0.23	6.80	0.00
Process production and departmental development	0.47	2.00	0.24	15.09	93.00	0.16	1.45	0.24
Process quality improvement and cost reduction	0.80	2.00	0.40	20.44	93.00	0.22	1.83	0.17
Resource deficiency and benefit uncertainty	0.52	2.00	0.26	7.76	49.00	0.16	1.64	0.20
Existence method competency and cost accumulation	0.31	2.00	0.15	15.10	49.00	0.31	0.50	0.61
Ambiguity and irrelevancy of SDMs	0.10	2.00	0.05	9.64	49.00	0.20	0.26	0.77
Process cost and time management	0.10	2.00	0.05	14.38	86.00	0.17	0.30	0.74
Process objectives and goals	1.32	2.00	0.66	19.65	86.00	0.23	2.89	0.06
System documentation, portability and maintainability	0.48	2.00	0.24	20.66	88.00	0.23	1.01	0.37
System success	0.88	2.00	0.44	15.45	87.00	0.18	2.49	0.09

Table 4.47: Lavene Test of Homogeneity - Job category

b) Analysis of variance of Job category

Consequently, ANOVA was employed in order to test for statistical significance between the means of “job category” variables; the variances between project managers, developers and other professionals against CSFs, SDM factors (quality and process), project success factors (product and system) and SDMs non-use reason factors were analyzed. It was observed that there is a significant difference between means of job categories (project managers, developers and other professionals) towards the factors such as “project complexity” critical success and the “system execution” success factor.

To further illustrate, the analysis shows that there was a significant difference of mean variances between project managers and other professionals (except systems developers) with regard to “project complexity” and “success execution” factors.

Analysis of variance: Marked effects are significant at p < .05000 Include condition: Job category - Project Manager, Developer & other

Variable	SS - Effect	df - Effect	MS - Effect	SS - Error	df - Error	MS – Error	F		Welch df - Effect	Welch df - Error	Welch F	Welch p
Standard methods	2.13	2.00	1.07	65.97	100.00	0.66	1.62	0.20	2.00	37.06	1.72	0.19
Project's goals and objectives	1.22	2.00	0.61	42.05	99.00	0.42	1.43	0.24	2.00	33.17	1.48	0.24
User-developer collaboration	2.32	2.00	1.16	39.99	99.00	0.40	2.88	0.06	2.00	44.20	3.10	0.06
User requirements and involvement	1.79	2.00	0.89	44.16	99.00	0.45	2.00	0.14	2.00	41.98	1.83	0.17
Technical adequacy and project milestones	0.97	2.00	0.49	28.68	99.00	0.29	1.68	0.19	2.00	38.67	1.71	0.19
Project plan and control	2.44	2.00	1.22	44.94	99.00	0.45	2.69	0.07	2.00	36.30	2.73	0.08
User training and system deployment	1.17	2.00	0.59	35.60	99.00	0.36	1.63	0.20	2.00	33.37	1.78	0.19
Developer's expertise	0.51	2.00	0.25	40.09	99.00	0.40	0.63	0.54	2.00	42.04	0.61	0.55
Project complexity	6.75	2.00	3.38	60.16	95.00	0.63	5.33	0.01	2.00	35.32	4.90	0.01
Project complexity & risk realization	1.98	2.00	0.99	51.37	99.00	0.52	1.90	0.15	2.00	34.99	2.08	0.14
Project development and completion	0.81	2.00	0.40	50.44	100.00	0.50	0.80	0.45	2.00	34.41	0.82	0.45
Conflicting requirement	0.75	2.00	0.38	76.74	99.00	0.78	0.48	0.62	2.00	35.22	0.45	0.64
Project resources	0.03	2.00	0.01	49.56	99.00	0.50	0.03	0.97	2.00	37.24	0.03	0.97
Top management Involvement	0.62	2.00	0.31	47.21	99.00	0.48	0.65	0.52	2.00	43.46	0.84	0.44
Systems development method impact	1.38	2.00	0.69	53.54	93.00	0.58	1.20	0.31	2.00	47.50	1.33	0.28
Process production and departmental development	2.40	2.00	1.20	37.79	93.00	0.41	2.95	0.06	2.00	28.97	3.05	0.06
Process quality improvement and cost reduction	2.21	2.00	1.11	60.00	93.00	0.65	1.72	0.19	2.00	24.44	1.76	0.19
Resource deficiency and benefit uncertainty	1.66	2.00	0.83	28.32	49.00	0.58	1.43	0.25	2.00	13.18	0.82	0.46
Existence method competency and cost accumulation	0.81	2.00	0.41	45.20	49.00	0.92	0.44	0.65	2.00	14.60	0.42	0.66
Ambiguity and irrelevancy of SDMs	1.40	2.00	0.70	38.35	49.00	0.78	0.89	0.42	2.00	14.05	0.80	0.47

Process cost and time management	0.68	2.00	0.34	50.15	86.00	0.58	0.58	0.56	2.00	30.97	0.62	0.55
Process objectives and goals	2.04	2.00	1.02	47.69	86.00	0.55	1.84	0.17	2.00	39.13	3.03	0.06
System documentation, portability and maintainability	1.25	2.00	0.63	41.20	88.00	0.47	1.34	0.27	2.00	35.69	1.56	0.22
System success	3.26	2.00	1.63	43.03	87.00	0.49	3.30	0.04	2.00	36.83	3.37	0.05

Table 4.48a: Analysis of variance - Job category

Unequal N HSD; variable: Project complexity: Marked differences are significant at p < .05000
Includes condition: Job category

Job category	Project manager **M=1.8235**	Developer **M=2.0000**	Other **M=2.6429**
Project manager		0.63	0.02
Developer	0.63		0.09
Other	0.02	0.09	

Table 4.48b: Analysis of variance - project complexity (Job category)

Unequal N HSD; Variable: system success: Marked differences are significant at p < .05000
Include condition: Job category

Job category	Project manager **M=2.0862**	Developer **M=2.5051**	Other **M=2.2708**
Project manager		0.07	0.80
Developer	0.07		0.69
Other	0.80	0.69	

Table 4.48c: Analysis of variance –system success (Job category)

c) Lavene Test of Homogeneity on business sectors

The views from different business sectors (system development, transport and communication and government) were examined in Section A7 of Chapter 4. Their variance homogeneities were tested and it was observed that sectors have a significantly different variance towards “system success” factors (refer to table 4.49a).

Lavene Test of Homogeneity of variances: Marked effects are significant at p < .05000 Include condition: Organisation- Systems development, Transport & communication & Banking								
Variable	SS - Effect	df - Effect	MS - Effect	SS - Error	df - Error	MS - Error	F	p
Standard methods	0.13	2.00	0.06	14.45	76.00	0.19	0.34	0.71
Project's goals and objectives	0.29	2.00	0.14	8.47	75.00	0.11	1.28	0.28
User-developer collaboration	0.15	2.00	0.08	10.83	75.00	0.14	0.53	0.59
User requirements and involvement	0.41	2.00	0.21	9.83	75.00	0.13	1.57	0.22
Technical adequacy and project milestones	0.10	2.00	0.05	7.49	75.00	0.10	0.50	0.61
Project plan and control	0.01	2.00	0.00	11.22	75.00	0.15	0.03	0.97
User training and system deployment	0.12	2.00	0.06	11.59	75.00	0.15	0.38	0.69
Developer's expertise	0.42	2.00	0.21	9.39	75.00	0.13	1.67	0.20
Project complexity	0.70	2.00	0.35	14.28	72.00	0.20	1.77	0.18
Project complexity & risk realization	0.28	2.00	0.14	17.88	75.00	0.24	0.59	0.56
Project development and completion	0.02	2.00	0.01	12.51	76.00	0.16	0.06	0.94
Conflicting requirement	1.16	2.00	0.58	15.23	75.00	0.20	2.87	0.06
Project resources	0.16	2.00	0.08	15.10	75.00	0.20	0.40	0.67
Top management Involvement	0.25	2.00	0.12	11.30	75.00	0.15	0.81	0.45
Systems development method impact	0.92	2.00	0.46	19.37	74.00	0.26	1.76	0.18
Process production and departmental development	0.34	2.00	0.17	12.59	74.00	0.17	1.01	0.37
Process quality improvement and cost reduction	0.19	2.00	0.10	15.37	74.00	0.21	0.46	0.63
Resource deficiency and benefit uncertainty	0.31	2.00	0.15	5.23	28.00	0.19	0.83	0.45
Existence method competency and cost	0.36	2.00	0.18	7.54	28.00	0.27	0.67	0.52

accumulation								
Ambiguity and irrelevancy of SDMs	1.06	2.00	0.53	5.37	28.00	0.19	2.77	0.08
Process cost and time management	0.60	2.00	0.30	9.61	63.00	0.15	1.97	0.15
Process objectives and goals	1.22	2.00	0.61	12.30	63.00	0.20	3.12	0.05
System documentation, portability and maintainability	0.69	2.00	0.34	17.53	65.00	0.27	1.27	0.29
System success	1.12	2.00	0.56	11.23	64.00	0.18	3.20	0.05

Table 4.49a: Lavene Test of Homogeneity - business sectors

7.3.3.5. Analysis of variance on business sectors

ANOVA was employed to test the statistical significance between the means of business sector variables. Practically, the analysis measures the variances of means between business sectors such as “systems development”, transport and communication and government towards CSFs, SDMs factors (quality and process), success factors (product and system) and SDMs non-use reason factors. It was observed that there was a significant difference between means of business sectors towards the “systems development method impact” and “project’s goals and objectives” success factors.

A particular observation was that there was a significant difference of mean variances of systems development business sectors to both transport and communication and government sectors with regard to “systems development method impact" and “process objectives and goal” success factors (refer to the table below).

Analysis of variance : Marked effects are significant at p < .05000 Include condition: Organisation - Systems Development, Transport & Communication & Banking

Variable	SS - Effect	df - Effect	MS - Effect	SS - Error	df - Error	MS - Error	F	P	Welch df - Effect	Welch df - Error	Welch F	Welch p
Standard methods	1.93	2.00	0.97	41.84	76.00	0.55	1.75	0.18	2.00	33.29	1.65	0.21
Project’s goals and objectives	0.42	2.00	0.21	30.17	75.00	0.40	0.53	0.59	2.00	35.54	0.53	0.59

User-developer collaboration	1.79	2.00	0.90	34.08	75.00	0.45	1.97	0.15	2.00	32.15	1.94	0.16
User requirements and involvement	0.18	2.00	0.09	31.45	75.00	0.42	0.22	0.80	2.00	31.18	0.26	0.78
Technical adequacy and project milestones	0.89	2.00	0.45	24.74	75.00	0.33	1.35	0.26	2.00	31.11	1.39	0.27
Project plan and control	2.57	2.00	1.28	39.62	75.00	0.53	2.43	0.09	2.00	32.67	2.35	0.11
User training and system deployment	0.03	2.00	0.01	30.40	75.00	0.41	0.03	0.97	2.00	34.74	0.04	0.96
Developer's expertise	0.37	2.00	0.18	25.90	75.00	0.35	0.53	0.59	2.00	30.65	0.46	0.64
Project complexity	1.67	2.00	0.84	50.12	72.00	0.70	1.20	0.31	2.00	27.80	1.06	0.36
Project complexity & risk realization	1.41	2.00	0.70	46.39	75.00	0.62	1.14	0.33	2.00	33.82	1.10	0.34
Project development and completion	0.20	2.00	0.10	38.22	76.00	0.50	0.20	0.82	2.00	32.06	0.18	0.83
Conflicting requirement	0.38	2.00	0.19	61.10	75.00	0.81	0.24	0.79	2.00	37.48	0.27	0.77
Project resources	1.82	2.00	0.91	38.73	75.00	0.52	1.76	0.18	2.00	29.87	1.41	0.26
Top management Involvement	0.56	2.00	0.28	36.54	75.00	0.49	0.57	0.57	2.00	31.83	0.42	0.66
Systems development method impact	4.13	2.00	2.07	41.65	74.00	0.56	3.67	0.03	2.00	32.16	5.56	0.01
Process production and departmental development	1.67	2.00	0.84	27.39	74.00	0.37	2.26	0.11	2.00	31.55	2.34	0.11
Process quality improvement	2.31	2.00	1.16	45.89	74.00	0.62	1.86	0.16	2.00	33.13	2.19	0.13

and cost reduction												
Resource deficiency and benefit uncertainty	0.58	2.00	0.29	17.23	28.00	0.62	0.47	0.63	2.00	16.49	0.59	0.56
Existence method competency and cost accumulation	1.13	2.00	0.57	25.71	28.00	0.92	0.62	0.55	2.00	16.02	0.79	0.47
Ambiguity and irrelevancy of SDMs	0.71	2.00	0.35	25.63	28.00	0.92	0.39	0.68	2.00	17.08	0.48	0.63
Process cost and time management	1.63	2.00	0.82	35.38	63.00	0.56	1.46	0.24	2.00	24.68	1.37	0.27
Process objectives and goals	5.63	2.00	2.81	27.33	63.00	0.43	6.49	0.00	2.00	27.47	4.09	0.03
System documentation, portability and maintainability	1.54	2.00	0.77	32.95	65.00	0.51	1.52	0.23	2.00	30.67	1.65	0.21
System success	1.14	2.00	0.57	32.34	64.00	0.51	1.13	0.33	2.00	27.21	0.96	0.40

Table 4.49b: Analysis of variance - business sectors

Unequal N HSD; variable: Systems development method impact: Marked differences are significant at p < .05000 Include condition: Business sector- System development, Transport & communication & Bank

Business sector	System Development	Transport and communication	Financial, Banking
Systems development		0.08	0.73
Transport and communication	0.08		0.29
Financial, Banking	0.73	0.29	

Table 4.49c: Analysis of variance- Systems development method impact (business sectors)

Unequal N HSD; variable: Process objectives and goals: Marked differences are significant at p < .05000 Include condition: Business sector - Systems development, Transport & communication & Banking			
Business sector	System Development **M=2.0068**	Transport and communication **M=2.7436**	Financial, Banking **M=2.0156**
Systems development		0.016084	0.999280
Transport and communication	0.016084		0.017616
Financial, Banking	0.999280	0.017616	

Table 4.49d: Analysis of variance - Process objectives and goals (business sectors)

This section was aimed at expressing the relationships and correlations between the CSFs, SDMs factors (quality and process), success factors (product and system) and SDMs non-use reason factors, using multiple regression through the exploitation of the statistical analyses such as multiple regression, correlation matrices, T-test, Anova and the Lavene Test of Homogeneity.

4.8 Conclusion

The chapter provides the preliminary findings of a questionnaire survey based upon prior research that attempts to investigate the relationship between systems development methodologies (SDMs) and information technology (IT) project success, and to investigate other critical success factors (CSFs) that contribute to information technology (IT) project success. The findings are statistically significant and we hope that these exploratory findings make a contribution to the study of the success of IT projects and the findings are also useful for further work. Quantitatively, the survey presents the results of the analysis of the following aspects: demographic information, critical success factors, the use and benefit of systems development methodologies, the reasons for not using systems development methodologies and project outcomes. Consequently, a T-test, ANOVA and Lavene Test of Homogeneity were exploited to measure and compare the intensity of the relationships between the variables.

The factors were observed for their relationships with the independent (systems development methodology critical success) and dependent (information technology project success) aspects. The primary analysis techniques embrace mean, frequency, factor analysis, multiple regression and correlation matrices with the depiction of graphs such as histogram, bar, pie and scatter points to emphasize the relationships and correlations. The study provides very good factor analyses figures and as such we can corroborate the supposition that the variables in this study provide a sense of consistency.

Chapter 5: Discussion

5.1 Introduction

The aim of this chapter is to provide a discussion of the statistical analyses obtained from the survey conducted in this study. The basis of discussion is the answers to the research questions formulated at the beginning of the study. The fundamental demographic results will also be summarized and interpreted. The chapter will subsequently identify and thrash out the limitations and suggest future work to give further innovative solutions with respect to the best practices of the deployment of IT projects. Lastly, we will review the reasons for not using SDMs and further discuss other discoveries mined from this study. Finally, the conclusion and recommendation will be presented based on the nature of study and the outcomes of the study.

5.2 The demographic information

The analyses show that the respondents were predominantly male (84%) with the female complement only 16%. Between the ages of 19-25yrs, male and female were 20% and 3% respectively, between 26-35yrs, male and female were 42% and 2% respectively, between 36-45yrs, male and female were 18% and 10% respectively and in the category of 46yrs or more, male and female were 4% and 2% respectively. Most of the respondents were systems developers. Results show that the three most prevalent qualifications would seem to be an honours degree, an undergraduate degree and a diploma respectively. Most of the respondents worked for organizations with more than 1000 employees. The most popular experiences in systems development were recognized to be 11yrs or more, 3-5yrs and 6-10yrs correspondingly. The size of most projects that respondents had been involved in were large and in line with completion periods which were subsequently gauged to be either less than one year, 1-2 years, 3-5 years and 6 years or more.

Furthermore, the answers to the research questions are provided. Each of the successive sections is devoted to one of the research questions.

5.3 The research questions

5.3.1. Does a relationship exist between systems development methodologies (SDMs) and information technology (IT) project success?

The multiple regression results show that SDMs do not predict IT project success (Refer to 4.7.2). However, there is a proven correlation between SDMs and IT project success. Moreover, 35% of the respondents indicated that they had used systems development methodologies in 76% or more projects they were involved in, and 27% of respondents indicated that 76% or more people in their organizations have used systems development methodologies. Consequently, the horizontal use of SDMs indicates that 34% of respondents agreed on the use of 76% or more.

From this evidence, we can assume that there is a positive influence of SDMs to the development of IT project success. To further strengthen this proof, we have in the survey listed the beneficial factors influenced by systems development methodologies practices and classified them into two aspects, which are quality and process factors.

5.3.1.1. Quality factors

The relationship between quality factors influenced by systems development methodologies and IT project success was measured (Refer to 4.4.7). The analyses confirm that most of the respondents agreed that all underlying factors influenced by systems development methodologies advanced the development of IT projects, and the most popular factor discovered was that systems development methodologies help to develop better systems.

In Table 4.22b, factor analyses were exercised to determine the strength of factors and we discovered that all factors were strongly equivalent to one another, thus showing a high sense of consistency. A single factor was constructed from these factors and it was named "*systems development method impact"* within which effectiveness and efficiently of systems development was conflated. Overall, an evaluation was done to test whether systems development methodologies:

- Help to develop more portable systems;
- Help to develop more efficient systems;
- Help to make users more satisfied with our systems;
- Help to develop more functional systems;
- Help to develop more reliable systems;
- Help to develop more maintainable systems;
- Help to develop more usable systems;
- Help to develop better systems.

5.3.1.2. Process factors

The relationship between the process factors influenced by systems development methodologies and IT project success was determined (Refer to 4.48). The analyses prove that most of the respondents agree that adopting the process of IT systems development methodologies in the development of a system is beneficial to systems development, and the most popular factor discovered was that systems development helps to improve the quality of the emerged systems.

In Table 4.24b, factor analysis was done to determine the strength of factors and we discovered two factors which were named "*process production and departmental development"* and "*process quality and cost management"*. Typically, it was observed that SDMs help to improve departmental productivity, quality and decrease cost. An evaluation was done to test whether systems development methodologies:

- Help to develop new applications faster;
- Help to develop an improvement of the functionality of new applications;
- Help to increase the productivity of the application developers;
- Help to develop a decrease in the cost of systems development;
- Help to develop an improvement in the quality of the systems;
- Help to develop a decrease in the cost of systems maintenance;
- Help to develop an improvement in the documentation of the systems;
- Help to improve the morale in my IS department;
- Help to achieve the goals of our IS department; and
- Help to improve our IS department's reputation for excellent work.

In summary: to answer the question, though there is no proof that SDMs should be one of the predictors to the IT project success, the analyses have proven that there is a positive impact and correlation between systems development methodologies (SDMs) and IT project success. We concur with Dyba *et al*. (2005:447) to the effect that systems development methodologies and their impacts are valuable and have a relationship with the performance of organizations.

5.3.2. To what extent do systems development methodologies (SDMs) influence information technology (IT) project success?

In Table 4.17a, though STRADIS seems to be a popular commercial method, it is clear that a high number of respondents who reported using it revealed that they only used it to some extent. The second popular commercial method is XP in which most of respondents indicated that they have used it to some extent, followed by the respondents who pointed out that they had never used it at all. Henceforth most of the underlying commercial methods such as IE, RUP and JMRAD, though they have been used to some extent, seem never to have been used by most respondents. On the other hand, YSM proved to have many respondents who did not use them at all, and in the next highest rating it was indicated that they have used it to a small extent.

The other point to make is that in both quality and process aspects that are influenced by systems development methodologies it emerges that most of the respondents do agree that these aspects have a positive impact to the development of information systems. Subsequently, for quality and process factors, a high percentage of respondents indicated a sense of Total Agreement and Neutral respectively towards this thought. Less response is observed on the use of in-house methods. In average use, the scrum and agile methods seemed to be engaged.

To answer the question, the analyses proved that systems development methodologies have been used to some extent and that most of the respondents have agreed that the aforementioned process and quality factors have improved the development of IT systems.

5.3.3. What other CSFs influence information technology project success?

The knowledge gained from the analyses was that the following critical success factors are observed to be among the top ten prominent factors that have an impact on the success of projects, and they are arranged in order of importance as per respondents' views: Refer to Table 4.11.

- A focus on the most important problems or opportunities;
- Senior management involvement;
- Use of appropriate hardware and software technologies;
- Good developers' technical expertise;
- Developers were familiar with the technology base;
- Developers' knowledge of the systems development process was good;
- Clearly defined system goals and objectives;
- Good developers' knowledge of the application domain;
- Users' expectations were real; and
- There was availability of the required technology and expertise to accomplish the specific technical action steps.

In Table 4.11a, factor analysis was done to determine the strength of factors (Refer to) and we discovered 13 factors which were named standard methods, project goals and objectives, user-developer collaboration, user requirements and involvement, technical adequacy and project milestones, project plan and control, user training and systems deployment, developers' expertise, project complexity (which is split between project complexity and project complexity and realization), project development and completion, conflicting requirements, project resources and top management involvement.

In Table 4.39, the correlation matrix analyses were done to measure the correlation between CSFs with IT project success factors which are classified into four facets as per the degree of their strength (Refer to Table 4.37a and Table 4.37b), namely:

- Process cost and time management;
- Process objectives and goals;
- System documentation, portability and maintainability; and

- System success.

It was discovered that CSFs such as conflicting requirements, project complexity, and project resources do not show a positive correlation to IT projects success, whereas CSFs like user requirements and involvement and developer's expertise partially show a positive correlation to IT project success. Otherwise, the rest of the underlying CSFs like standard methods, project goals and objectives, user-developer collaboration, technical adequacy and project milestones, project plan and control, user training and system deployment, project complexity and risk realization, project development and completion, and top management involvement show a significant positive correlation to IT project success factors.

Furthermore, multiple regression analyses were utilized to determine whether each variable is responsible for or dependent on the success of IT project factors, and the results prove that CSFs such as project complexity and risk realization, technical adequacy and project milestones, user training and system deployment, project plan and control, user-developer collaboration and user training and system deployment are observed to be the predictors of IT project success factors. Refer to 4.7.2.

To compare these CSFs with the ones from the literature study and results of the Chaos Report 2009, it emerged that senior management involvement, adequate hardware and software use and systems development methodologies knowledge and clear goals and objectives were recognized in the literature review, whereas in the Chaos Report 2009, senior management involvement, hardware and software technologies, clear business goals and objectives and skilled resources were recognized. Certainly, from this study, these CSFs are still gaining in popularity.

5.3.4. What effects do systems development methodologies have as compared to other factors contributing to information technology (IT) project success?

5.3.4.1. Systems development methodologies

Systems development methodology is deemed to improve quality in the development of systems (Yahya *et al.,* 2002:15). In this dissertation the quality factors influenced by the practices of SDMs were assumed to achieve the following aspects (Refer to 4.22b)**:**

- Improved systems;
- Usable systems;
- Maintainable systems;
- Reliable systems;
- Functional systems;
- User satisfaction;
- Efficient systems; and
- Portable systems.

In Table 4.22b, factor analyses prove that these aspects are parallel to each other; hence they are assumed to be effective in the development of the system as per analysis response. They were collectively named “the systems development method impact”.

Furthermore, SDMs and process models are assumed to guide IT developers to advance the development systems (Dyba *et al.,* 2005:449). The process was assessed and inferred to achieve the following characteristics. (Refer to Table 4.24a);

- Quality improvement;
- Reputation improvement.
- Goal achievement;
- Functionality improvement;
- Productivity increases;
- Faster application development;
- Maintenance cost decreases;
- Morale improvement;
- Development cost increases; and
- Documentation improvement.

In Table 4.24b, the factor analysis was done to determine the strength of association among these factors and two factors were discovered to classify them – “*process* *production and departmental development"* and "*process quality and cost management".*

Additionally, the SDMs factor as one of the CSFs does not predict any of the following IT project success factors vs. process cost and time management, process objectives and goals, system success and system documentation, portability and maintainability. (Refer to 4.7.24). Nonetheless, SDMs signify a correlation with all IT project success factors except the system success factor.

5.3.4.2. Critical success factors

Through the employment of factor analysis, 13 factors were extracted to constitute the raw critical success factors, and they can be named as: standard methods, project goals and objectives, user-developer collaboration, user requirements and involvement, technical adequacy and project milestones, project plan and control, user training and system deployment, developers' expertise, project complexity (which is split between project complexity and project complexity and realization), project development and completion, conflicting requirements, project resources and top management involvement. (Refer to Table 4.11a).

The aforementioned factors were measured against the project success factors such as process cost and time management, process objectives and goals, system success and system documentation, portability and maintainability (Refer to 4.7.2). It was also realized that the project complexity and risk realization made a significant contribution to IT project success. Conversely, on average CSFs show a positive linear correlation with IT projects success.

To answer the question on how to compare SDMs and other CSFs, systems development methodologies are judged to be positive and critical to project success since they entail the process and procedures that improve the systems development quality and effectiveness which may also be operationalised by executing the underlying critical success factors. Masrek *et al*. (2008:137) indicated that the techniques of SDMs assist practitioners in planning, managing control and estimating the information systems projects

In contrast, the independent critical success factors cater for autonomous purposes which may fail to articulate various valuable attributes that may less positively impact on the development of information systems. We have learnt from the assumption by Dalcher and Brodie (2007:273) that the success factor is the endeavour to predict the good practices needed to accomplish the project successfully. Though the lack of success factors does not ensure a failure, yet it can be perceived as a phenomenon that can jeopardize the project. As a result, we perceived them and their impacts to have less buoyancy to achieve the success of the project.

Certainly, the technique of SDMs is packaged along with the process, method, people's view and tools and techniques which are believed to impact IT projects positively; as per the study by Huisman and Iivari (2006:32), SDM comprises a philosophical approach, process model, method, and tools and techniques, whereas, except in the application of a batch of CSFs, a single success factor is unlikely to impact IT projects positively.

5.3.5. Which methodology is mostly used in organisations?

It emerged from the analyses that STRADIS is the most popular SDM used in organizations. It attained the highest popularity among other SDMs, thus it predominantly rates a high percentage within a context of being used to some extent. Subsequently, the popularity points to XP which has a high number of respondents who signify that they use it to some extent; otherwise, the respondents show that they did not use IE, RUP, JMRAD and YSM respectively. We have also gained the knowledge that the three in-house or custom methods that have been used the most in the organisations were SCRUM which seems to be used to a greater extent, followed by AGILE and PMBOK respectively. Refer to Table 4.18a and Table 4.18b.

To answer the question, STRADIS is believed to be the most popularly used commercial SDM and the SCRUM demonstrated a high level of favour among customized methods.

The research questions which constitute the aims and goals of this study were addressed and discussed comprehensively. The following subsection will provide other fundamental innovations from this study. The aspects, namely reason for not using SDMs and the different perceptions of agile and traditional SDMs, gender, jobs and business sectors will be expressed. Finally, limitations and future work for this study will be presumed.

5.4. The reasons for not using a systems development methodology

In Figure 4.31, out of 132 respondents, only 41 respondents disclosed their experience of not using SDMs. 19 respondents indicated that they had considered using them but decided otherwise, 12 claimed to have used them but abandoned them and 10 confessed to never having used them at all. The most popular reason for not using SDMs was that the profile of development projects in their IS department did not require the use of systems development methods.

Furthermore, In Table 4.33, factor analysis was used and the non-use factors were classified into three groups of factors which were named resource deficiency and benefit, traditional method adequacy, ambiguity and irrelevance of system development methods uncertainty in adopting systems development method.

Though the analyses confirm a certain level of adaptation of commercial SDMs, the highest percentages designate that they did not undergone changes when they were implemented; which means that most people did not change the system methodologies when they were used. Refer to Table 4.18a

5.5. The perceptions of respondents towards SDMs, CSFs and IT project success factors

The section compares the perceptions of categories such as agile and traditional SDMs, genders, job category and business sectors in relation to CSFs, SDMs factors (quality and process), success factors (product and system) and SDMs non-use reason factors.

It was learnt from Table 4.44 and Table 4.45 that there are different impacts caused by traditional and agile methods towards the factors namely project technique and developers' proficiency, level of complexity and risk realization, process quality improvement and cost reduction and system success. Furthermore, the results suggest that factors such as process quality improvement and cost reduction made a great contribution to the different adaptations made to traditional and agile method application.

In Table 4.46, the results suggested that perceptions of females and males were contrasting in terms of the following CSFs namely user-developer collaboration, user requirements and involvement, top management involvement, resource deficiency and benefit uncertainty and ambiguity and irrelevancy of SDMs.

It is believed that the views of project developers and project managers and other job categories experience dissimilarity with regard to the CSF like project complexity and success factor like system success (Refer to Table 4.47 and Table 4.48a). Furthermore, their opinions are extensively different with regard to factors such as user-developer collaboration and systems development method impact.

In Table 4.49a and Table 4.49b, the results show that sectors such as systems development, transport and communication and banking have different experiences with regard to systems development method impact and success factors like process objectives and goals, and their interactions with success factors like system success are comprehensively different.

5.6. Limitations and future work

- Some limitations exist in that the impact of method-in-action to project success was not fully studied. It can be a good topic for future research. This concept is underpinned by Fitzgerald (2002).
- Another direction for future studies can be in-depth knowledge about the perception of IT developers and managers. The study conducted by Huisman and Iivery (2006:39) indicated that IT developers and managers acquire different opinions regarding SDMs and their impacts. Further review of the impact of high experienced and less skilled practitioners about the adoption of systems development methodologies (SDMs) should be valuable. The study by Omland and Nielsen (2009:216) assumed that the more experienced developers perceived methods and their impact as less important.

- The nature and size of projects utilizing systems development methodologies practices should be investigated more precisely to determine how they affect IT project success.
- The study results have failed to find evidence as to whether systems development methodologies have a direct impact on the success of IT projects. As a result, future studies could measure the impact of particular SDMs. Specifically, according to Chow and Cao (2008:961) research into agile methods is still limited in academics circles.
- The results of this study suggest a need to discover the realities involved in the usage and substantiation of methodologies with regard to IT project success in future. Dyba *et al.* (2005:447) conducted a study to measure methodology usage and this concept needs to be further examined.
- Future research should scrutinize the concept that employment of SDMs practices eliminates or carries away the authorities and responsibilities from the practitioners. According to Fitzgerald *et al.* (2002:124) JSD was aimed at minimizing the responsibility of the developer in the development process.
- Future studies should seek to find out the risks associated with the adoption of SDMs. Duggan and Richest (2006: xv) point out that the deployment of SDMs is associated with some risks.
- The study could have obtained richer and more insightful information regarding SDMs and critical success factors if more data had been collected in the organizations.
- If possible, similar research should be conducted periodically because it provides knowledge to the practitioners who are interested in the domain of software development and will make them aware of potential problems and proposed solutions.

5.7. Conclusion

This chapter was aimed at presenting the knowledge gained from the statistical analysis provided in Chapter 4. We provided answers to the research questions which were considered to be the main objectives of this study. The demographic information and the reasons for not using SDMs results were in supplementary fashion summarized and interpreted.

It was found that, to some extent, there is a relationship between systems development methodologies (as we saw, high numbers of respondents agree that those quality and process factors influenced by SDMs improve the development of IT projects). Most of these factors have a positive correlation with IT project success. Nonetheless, an SDM factor is not dependent on or responsible for the success of IT projects. A high number of practitioners and projects utilize the practice of SDMs. On the other hand, it is recognized that for reasons such as competency of current method, SDMs were not used, and to some degree, SDMs were adapted to fit business needs.

The top ten critical success factors which contribute to IT project success were also investigated and outlined. It was indicated that the most popular factor was the identification of potential benefit and risks.

On average, the study confirms that there is a relationship between systems development methodologies (SDMs) and IT project success. We concur with Dyba *et al*. (2005: 447) that systems development methodologies and their impacts are valuable and have a direct relationship with the performance of organizations.

In conclusion, the study has hopefully provided a better understanding of the relationships between SDMs and IT project success. Predominantly, as per previous studies, the organization would now have a better understanding of the potential benefits and advantages that SDMs can offer to their information systems development pursuit. Conversely, they can also enhance knowledge about the probable constraints and limitations that go along with the adoption of SDMs in attempt of developing information systems. Moreover, the study enhances the reader's knowledge about effective critical success factors that influence the success of the deployment of IT projects. Particularly, the findings of the research should be of most interest to practitioners and academics in the information systems and software development domain.

We recommend that studies about the conceptualization and operationalisation of SDMs need to be conducted to equip practitioners with knowledge in their endeavour to develop IT projects; since the study discloses that, for some stipulated reasons, there are still practitioners who do not employ SDM practices. The other fact is that the practices of SDMs are always associated with the development of systems (Duggan & Reichgelt, 2006:9), hence it is vital to dig deeper regarding the concept of SDMs. The other main reason will be to address the underlying limitations of this study and follow up suggestions for future work.

References

Akmanligil, M. & Palvia, P.C. 2004. Strategies for global information systems development. *Information & management,* 42:45-59*.*

Al Neimat, T. 2005. Why IT projects fail. (The Project Perfect White Paper Collection). http://www.projectperfect.com.au Date of access: 02 July 2011.

Al-Mudimigh, S., Ullah, Z. & Alsubaie, T. 2010. A framework for portal implementation: a case for Saudi organizations. *International journal of information management ,* 31(1):38-43.

Ateneo de Mille University. Professional Schools Library. 2005. Information resources guide on methods of research. http://www.apslibrary.ateneo.edu. Date of access: 02 July 2011.

Avison, D. & Fitzgerald, G. 2008. Information systems development: methodologies, techniques & tools. 4th ed. London: McGraw-Hill. 35 p.

Barry, C. & Lang, M. 2003. A comparison of 'traditional' and multimedia information systems development practices*. Information and software technology,* 45:217-2*22.*

Bechina, A.A.A. & Ndlela, M.N. 2009. Success factors in implementing knowledge based systems. *Electronic journal of knowledge management,* 7(2):211-218.

Belout, A. & Gauvreau, C. 2004. Factors influencing project success: the impact of human resource management*. International journal of project management*, 22:1-11*.*

Benediktsson, O. 2002. Failed software development projects lessons to be learned. (Unpublished lecture at Glasgow Caledonian University, 4 February 2002).

Bern, A., Nikula, U., Pasi, S. & Smolander, K. 2007. Contextual factors affecting the software development process: an initial view. (2nd AIS SIGSAND European Symposium on Systems Analysis and Design. Gdansk, Poland, 5 June 2007).

Biehl, M. 2007. Success factor for implementing global information systems. *Communication of the ACM,* 50(1):52-58, Jan.

Blindenbach-Driessen, F. & Van den Ende, J. 2006. Innovation in project-based firms: the context dependency of success factors. *Research policy,* 35:545-561*.*

Brodie, L. & Dalcher, D. 2007. Successful IT projects. London: Middlesex University Press.

Cheng, T.C.E. & Chiu, I.S.F. 2008. Critical success factors of business process re-engineering in the banking industry. *Knowledge and process management,* 15(4):258-269.

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

Creswell, J.W. 2003. Research design: qualitative, quantitative, and mixed methods approaches. 2nd ed. Thousand Oaks: Sage.

Dahiya, D. & Jain, P. 2010. Enterprise systems development: impact of various software development methodologies. *International journal of advancements in computing technology,* 2(4):78-87, Oct.

Dalcher, D. & Brodie, L. 2007. Success IT projects. London: Thomas Learning.

Doi, R. 2006. Why IT projects fail. (Paper for the Fifth Audit Seminar of the INTOSAI Standing Committee on IT Audit, November 2006).

Dong, C., Chuah, K.B. & Zhai, L. 2004. A study of critical success factor of information system projects in China. (Proceedings of PMI Research Conference, London).

Duggan, E.W. & Reichelt, H., *eds*. 2006. Measuring information systems delivery quality. London: Idea Group Publishing.

Dyba, T., Moe, N.B. & Arisholm, M.E. 2005. Measuring software methodology usage: challenges of conceptualization and operationalization. (International Symposium on Empirical Software Engineering, 17-18 November).

Ehie, I. & Madsen, M. 2005. Identifying critical issues in enterprise resource planning (ERP) implementation. *Computer industry*, 56:545-557.

Esteves, J.M. & Pastor, J.A. 2002. Understanding the ERP project champion role and its criticality. (ECIS 2002 • June 6-8, Gdańsk, Poland).

Fitzgerald, B., Russo, N.L. & Stolterman, E. 2002. Information systems development: methods in action. London: McGraw-Hill. 194 p.

Fowler, M. 2001. The new methodology. *Journal of natural sciences,* 6(1-2):12-24, Dec.

Futrell, R.T., Shafer, D.F. & Shafer, L.I. 2002. Quality software project management. Upper Saddle River, NJ: Prentice Hall. 512 p.

Gai, S. & Xu, C. 2009. Research of critical success factors for implementing knowledge management in China. (International Conference on Information Management, Innovation Management and Industrial Engineering, 26-27 December. v. 4:562-564).

Griffin, A.S. & Brandyberry, A.A. 2008. System development methodology usage in industry: a review and analysis. *Journal of information system applied research (JISAR)*, 3(19):1-18.

Hughes, B. & Cotterell, M. 2009. Software project management. 5th ed. London: McGraw-Hill.

Huisman, H.M. 2000, .The deployment of systems development methodologies: A South African Experience., PhD Thesis, Potchefstroomse Universiteit vir CHO, Potchefstroom, South Africa, 242 pages.

Huisman, H.M. 2009. Information system engineering, 2. Potchefstroom: North-West University. (Study guide ITRW 624 E).

Huisman, H. & Iivari, J. 2006. Deployment of systems development methodologies: perceptual congruence between IS managers and systems developers. *Information & management,* 43:29-49.

Huisman, H & Iivari, J. 2007. The relationship between organisational culture and the deployment of systems development methodologies, *MISQ,* 31: 35-58. March

Humphrey, W.S. 2005. Why big software projects fail: the 12 key questions. *Crosstalk, the journal of defence software engineering:* 25-29, March.

Iivari, J., Hirschheim, R. & Klein, H.K. 2000. A dynamic framework for classifying information systems development methodologies and approaches. *Journal of management information systems,* 17(3):179-218.

Jackson, P. & Klobas, J. 2008. Building knowledge in projects: a practical application of social constructivism to information systems development. *International journal of project management,* 26:329-333.

Jarrar, Y.F., Yasar, F., Al-Mudimigh, A. & Zairi, M. 2010. ERP implementation critical success factors: the role and impact of business process management. (Research paper RP-ECBPM/0010). http://www.ecbpm.com Date of access: 02 July 2011.

Johnson, D.J. 2000. Levels of success in implementing information technologies. *Innovative higher education,* 25(1):59-76, Fall.

Johnson, R.B. & Onwuegbuzie, A.J. 2004. Mixed methods research: a research paradigm whose time has come. *Educational researcher,* 33(7):14-26.

Karlsen, J. T., Andersen, J., Birkely, L.S. & Ødeg Ard, E. 2005. What characterizes successful IT projects? *International journal of information technology & decision making,* 49(4):525-540.

Kelly, D., Harper, D. J. & Landau, B. 2008. Questionnaire mode effects in interactive information retrieval experiments. *Information processing and management,* 44:122-141.

Kerzner, H. 2006. Project management best practices: achieving global experience. Hoboken, NJ: Wiley. 158 p.

Kiely, G. & Fitzgerald, B. 2005. An investigation of the use of methods within information systems development projects. *Electronic journal of information systems in developing countries (EJISDC),* 22(4):1-13.

Kim, S. 2003. Research paradigms in organizational learning and performance: competing modes of inquiry. *Information technology, learning, and performance journal,* 21(1):9-18, Spring.

Kimble, C. 2008. Semi-formal methodologies: the first generation methodology. University of York.

Kotsonis, A. 2007. Evaluation of different approaches for the development of an information system. The coursework submission system of the Department of Information Studies project. Sheffield: University of Sheffield. (Dissertation - MSc.).

Krauss, S.E. 2005. Research paradigms and meaning making: a primer. *Qualitative report,* 10(4):758-770, Dec.

Kronbichler, S.A., Ostermann, H. & Staudinger, R. 2009. A review of critical success factors for ERP-projects. *Open information systems journal,* 3:14-25.

Lancaster University. Management School. 2005. Case study of successful complex IT projects. Sponsored by British Computer Society in association with Vehicle & Operator Service agency, August 2005.

Lang, R.D. 2007. Project leadership: key elements and critical success factors for IT project managers. *Journal of healthcare information management,* 21(1):2-5.

Laudon, K.C. & Laudon, J.P. 2000. Management information systems: organization and technology in the networked enterprise. 6th ed. Upper Saddle River, NJ: Prentice Hall.

Lee, C. & Lee, H. 2001. Factors affecting enterprise resource planning systems implementation in a higher education institution. *Issues in informatics systems:* 207-212.

Leem, C.S. & Kim, S. 2002. Introduction to an integrated methodology for development and implementation of enterprise information systems. *Journal of systems and software,* 60:249-261.

Lind, M.R. & Culler, E. 2009. The relationship between information technology critical success factors and project performance. (Proceedings ISECON/CONISAR, Washington, DC).

Lu, X., Xin, Z. & Han, H. 2008. Analysis on factors impacting to information system development. (2008 International Seminar on Future Information Technology and Management Engineering).

Mackenzie, N. & Knipe, S. 2006. Research dilemmas: paradigms, methods and methodology. *Issues in educational research,* 16. http://www.iier.org.au/iier16/ mackenzie.html Date of access: Date of access: 02 July 2011.

Madsen, S., Kaut, K. & Vidgen, R. 2006. A framework for understanding how a unique and local IS development method emerges in practice. *European journal of information systems,* 15:225-238.

Madsen, S. & Vidgen, R. 2009. A pragmatic approach to IS development and socio-technical evaluation. (17th European Conference on Information Systems).

Masrek, M., Hussin, N. & Tarmuchi, N. 2008. An exploratory study on systems development methodologies for web-based applications. *Information management & computer security,* 16(2):137-149.

Matveev, A.V. 2002. The advantages of employing quantitative and qualitative methods in intercultural research: practical implications from the study of the perceptions of intercultural communication competence by American and Russian managers. *Institute of Management, Business and Law Publishing*: 59-67.

McKelvey, B. 2002. Postmodernism vs. truth in management theory. (*In* Lock, E., *ed*. Post-modernism & management: pros, cons, and alternatives. Amsterdam: Elsevier. p. 113-168).

Mertens, D.M. 2005a. Research methods in education and psychology: integrating diversity with quantitative and qualitative approaches. 2nd ed. Thousand Oaks: Sage.

Mertens, D.M. 2005b. The transformative prism: resilience and social justice in mixed methods. (First International Congress of Qualitative Inquiry, University of Illinois, Urbana-Champaign).

Meso, P., Madey, G., Troutt, M.D. & Liegle, J. 2006. The knowledge management efficacy of matching information systems development methodologies with application characteristics: an experimental study. *Journal of systems and software,* 79:15-28.

Middleton, P. & McCollum, B. 2001. Management of process improvement by prescription. *Journal of systems and softwares,* 5:9-19.

Mihailescu, D. & Mihailescu, M. 2009. Exploring the nature of information systems development methodology: a synthesized view based on a literature review. (Proceedings of the Fifteenth Americas Conference on Information Systems, San Francisco, California August 6th-9th 2009 by an authorized administrator of AIS Electronic Library (AISeL)).

Moohebat, M.R., Jazi, M.D. & Asemi, A. 2010. A comparative study of critical success factors (CSFs) in implementation of ERP in developed and developing countries. *International journal of advancements in computing technology,* 2(5):99-110, Dec.

Morisio, M., Egorova, E. & Torchiano, M. 2007. Why software projects fail? Empirical evidence and relevant metrics. (Proceedings of the IWSM - Mensura).

Nah, F.F., Lau, J.L. & Kuang, J. 2001. Critical factors for successful implementation of enterprise systems. *Business process management journal,* 7(3):285-296.

Nandhakumar, J. &. Avison, D.E. 1999. The fiction of methodological development: a field study of information systems development. *Information technology & people,* 12(2):176-191.

Nasir, M.H.N. & Sahibuddin, S. 2011. Critical success factors for software projects: a comparative study. Scientific research and essays, 6(10):2174-2186, 18 May.

Ngai, E.W.T., Law, C.C.H. & Wat, F.K.T. 2008. Examining the critical success factors in the adoption of enterprise resource planning. *Computers in industry,* 59(6):548-564.

Oates, B.J. 2008. Researching information systems and computing. London: Sage.

Olson, D.L. & Zhao, F. 2007. CIOs' perspectives of critical success factors in ERP upgrade projects. *Enterprise information systems,* 1(1):129-138, Feb.

Omland, H.O. & Nielsen, P.A. 2009. Actors' competencies or methods? A case study of successful information systems development. (20th Australasian Conference on Information Systems. 2-4 Dec 2009, Melbourne).

Pieterse, P.J. 2006. An interpretive analysis of systems development methodology adaptation in South Africa. Potchefstroom: North-West University. (Dissertation - MSc.).

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

Project Management Institute. 2004. A guide to the project management body of knowledge. 3rd ed. (PMBOK guide). Newtown Square. 8 p.

Qumer, A. & Henderson-Sellers, B. 2008. An evaluation of the degree of agility in six agile methods and its applicability for method engineering. *Information and software technology,* 50:280-295.

Rainer, A. & Hall, T. 2002. Key success factors for implementing software process improvement: a maturity-based analysis. *Journal of systems and software,* 62:71-84.

Remus, U. 2006. Critical success factors of implementing enterprise portals: a comparison with ERP implementation. (Proceedings of the 39th Hawaii International Conference on System Sciences). *Business process management journal,* 13(4):538-582.

Rodriguez-Repiso, L., Setchi, R. & Salmeron, J. 2007. Modelling IT projects success: emerging methodologies reviewed. *Technovation,* 27:582-594.

Ross, K.N. 2005. Quantitative research methods in educational planning. Paris: UNESCO International Institute for Educational Planning. Available online: http://www.unesco.org/iiep.

Rowlands, B. 2004. The rational and political roles of methods in information systems development. (Proceedings Association for Information Systems AIS Electronic Library (AISeL), Paper 95).

Saini, H., Saini, D. & Gupta, N. 2009. E-Business system development: review on methods, design factors, techniques and tools with an extensive case study for secure online retail selling industry. *Indian journal of science and technology,* 2(5):82-90.

Sale, J.E.M., Lohfeld, L.H. & Brazil, K. 2002. Revisiting the quantitative-qualitative debate: implications for mixed-methods research. *Quality & quantity,* 36:43-53.

Sarker, S. & Lee, A. 2003. Using a case study to test the role of three key social enablers in ERP implementation. *Information & management*, 40:813-829.

Schwalbe, K. 2007. Information technology project management. 5th ed. Boston, Mass.: Thomson/Course Technology.

Serra, J.S. 2002. Participation in information systems development: problems and possible solutions. *COM331 SAAM. Participation in ISD.*

Siau, K. & Tan, X. 2005. Evaluation criteria for information systems development methodologies. (Proceedings of the Eleventh Americas Conference on Information Systems, Omaha, NE, USA August 11th-14th 2005. p. 31-32).

Sommerville, I. 2007. Software engineering. 8th ed. Boston: Addison-Wesley.

Standish Group International. 1995. Standish group report. Chaos, T23E-T10E Boston: The Standish Group International. http://www.scs.carleton.ca/~beau/PM/ Standish-Report.html. Date of access: 18 January 2011.

Standish Group International. 1996. Chaos reports. Unfinished voyages: a follow-up to the CHAOS report. Boston: The Standish Group International.

Standish Group International. 2001. Extreme chaos. Boston: The Standish Group International.

Standish Group International. 2005. Extreme chaos. Boston: The Standish Group International.

Standish Group International. 2009. Chaos, Summary. The 10 Laws of CHAOS. Report. Boston: The Standish Group International. http://www1.standishgroup.com/ newsroom/ chaos_2009.php Date of access: 8 January 2011.

Standish Group International. 2011a. Chaos reports. Boston: The Standish Group International. http://www.standishgroup.com

Standish Group International. 2011b. Chaos success factors. *Chaos activity news,* 6(4), April. http://www.standishgroup.com

Strode, D.E. & Huff, S.L. 2009. The impact of organizational culture on agile method use. (Proceedings of the 42nd Hawaii International Conference on System Sciences).

Sun, A.Y.T., Yazdani, A. & Overend, J.D. 2005. Achievement assessment for enterprise resource planning (ERP) system implementations based on critical success factors (CSFs). *International journal production economics,* 98:189-203.

Tan, W., Cater-Steel, A. & Toleman, M. 2009. Implementing IT service management: a case study focusing on critical success factors. *Journal of computer information systems,* Winter.

Tesch, D., Sobol, M.G., Klein, G. & Jiang, J.J. 2009. User and developer common knowledge: effect on the success of information system development projects. *International journal of project management,* 27:657-664.

Thomas, G. & Fernandez, W. 2008. Success in IT projects: a matter of definition? *International journal of project management,* 26:733-742.

Toleman, M., Almeida, A. & Darroch, F. 2004. Aligning adoption theory with agile system development methodologies. (Pacific Asia Conference on Information Systems 2004 Proceedings).

Truex, D., Baskerville, R. & Travis, J. 2000. A methodical systems development: the deferred meaning of systems development methods. *Accounting management & information technologies,* 10:53-79.

Turban, E., Mclean, E. & Wetherbe, J. 2001. Information technology for management: making connections for strategic advantage. New York: Wiley. 530 p.

Venable, J.R. & Lim, F.C.B. 2001. Development activities and methodology usage by Australian Web Site Consultants. (4th Western Australian Workshop on Information Systems Research (WAWISR 2001) organized by School of Information Systems Curtin University of Technology, Australia. p. 1-7).

Vidgen, R., Avison, D., Wood, B. & Wood-Harper, T. 2002. Developing web information systems. Oxford: Elsevier Butterworth-Heinemann.

Warwick, J. 2008. A case study using soft systems methodology in the evolution of a mathematics module. *Montana mathematics enthusiast* (*TMME),* 5(2&3):26.

Wateridge, J. 2005. IT projects: a basis for success. *Journal of project management,* 13(3):169-172.

Weaver, P. 2004. Success in your project: a guide to student system development projects. Upper Saddle Hill, NJ: Prentice Hall.

Westfall, R., Rainer, K. & Lai, L. 2001. Information technology for management. 2nd ed. New York: Wiley.

Wetherbe, T.M. 2001. Information technology for management. 2nd ed. Hoboken, NJ: Wiley.

Wixoma, B.H. & Watso, H.J. 2001. An empirical investigation of the factors affecting data warehousing success. *MIS quarterly*, 25(1):17-41, March.

Wong, K.Y. 2005. Critical success factors for implementing knowledge management in small and medium enterprises. *Industrial management & data systems,* 105(3):261-279.

Wourms, B. 2004. What makes a good IT project manager: five keys for success. pm solutions. http://www.pmsolutions.com. Date of access: 12 July 2011.

Xuhong, Y., Lio, K. & Qing, X. 2008. The influence of management support on enterprise's information system (IS): successful implementation: based on methods of field interview. (4th International Conference on Wireless Communications, Networking and Mobile Computing, WiCom '08. 4 p.).

Yaghini, M., Bourouni, A. & Amiri, R.H. 2009 . A framework for selection of information systems development methodologies. *Computer and information science,* 2(3):1.

Yahya, Y., Yusof, M.M., Yusof, M. & Omar, N. 2002. The use of information system development methodology in Malaysia. *Jurnal Antarabangsa (Teknologi Maklumat),* 2:15-34.

Yardley, D. 2002. Successful IT project delivery: learning the lessons of project failure. Boston: Addison-Wesley. 346 p.

Yeo, K.T. 2002. Critical failure factors in information system projects. *International journal of project management,* 20:241-246.

Yingjie, J. 2005. Critical success factors in ERP implementation in Finland. (Dissertation - M.Sc.). Swedish School of Economics and Business Administration.

Young, R. & Jordan, E. 2008. Top management support: Mantra or necessity? *International journal of project management,* 26:713-725.

Zwirowicz, A. 2009. The role of IE methodology in SDIs. (International Workshop on Spatial Information for Sustainable Management of Urban AreasFIG Commission 3 Workshop 2009. Mainz, Germany, 2-4 February 2009).

APPENDIX: Survey form

Research Questionnaire: The relationship between the system development methodologies (SDMs) and information technology (IT) project success

……

The purpose of this study is to bring the knowledge about the relationship between the systems development methodologies (SDMs) and Information Technology (IT) project success. The study also seeks to find other effective success factors that influence IT project success.

Your participation in this study is most appreciative and rest assured that your answers will be kept confidential and the completion of this questionnaire is voluntary.

The results of this study will be provided on request.

Return deadline: 15 March 2011

……

SECTION A: Demographic Information

1. Please indicate your gender.	
1.1. Male 1.2. Female	1 2

2. Please indicate your age.	
2.1. 18 or less 2.2. 19 – 25 2.3. 26 – 35 2.4. 36 – 45 2.5. 46 or more	1 2 3 4 5

3. Please indicate your job category.	
3.1. Project Manager/Leader 3.2. Developer 3.3. Other, please specify ……………………………………………………………… ………………………………………………………………	1 2 3

4. Please indicate your qualification	
4.1. Senior Certificate (High School) 4.2. Certificate or Diploma 4.3. University or Technicon degree 4.4. Honours degree 4.5. Masters' degree 4.6. Doctoral degree 4.7. Other, please specify …………………..	1 2 3 4 5 6 7

5. Please indicate the sector of your organization.	
5.1. Computer Vendor 5.2. Mining 5.3. Manufacturing 5.4. Electricity and water supply 5.5. Consulting 5.6. Tertiary sector 5.7. System Development 5.8. Transport and communication 5.9. Community services 5.10. Financial, Banking 5.11. Government 5.12. Other, please specify	1 2 3 4 5 6 7 8 9 10 11 12
6. Please indicate the size of your organization.	
6.1. 1–50 employees 6.2. 51–200 employees 6.3. 201 – 500 employees 6.4. 501 – 1000 employees 6.5. More than 1000 employees	1 2 3 4 5

7. Please indicate the province in which your organization is based.	
7.1. Gauteng 7.2. Free State 7.3. North West 7.4. Mpumalanga 7.5. Limpopo 7.6. Kwazulu Natal 7.7. Eastern Cape 7.8. Western Cape 7.9. Other, please specify...	1 2 3 4 5 6 7 8 9

8. Please indicate your personal experience in systems development.	
8.1. None	1
8.2. Less than 1 years	2
8.3. 1 – 2 years	3
8.4. 3 – 5 years	4
8.5. 6 – 10 year	5
8.6. 11 years or more	6

9. Please indicate the size of the last information systems development project you were involved with.	
9.1. Small	1
9.2. Medium	2
9.3. Large	3
9.4. Very Large	4
9.5. I have not participated in any information systems development project	5

10. Please indicate how long it took to complete the last project you were involved with.	
10.1. Less than a year	1
10.2. 1 – 2 years	2
10.3. 3 – 5 years	3
10.4. 6 years or more	4

SECTION B: Success Factors influencing Information Systems development

On the lines below, place an X to indicate the factor/s used successfully in developing the system in your department:

To what extents do you agree with the following statements as valid descriptions of the last systems development project you were involved with?	Not at All	To a little extent	To some extent	To a greater extent
1 The project used an appropriate systems development methodology.	1	2	3	4
2. The project had clear defined system goals and objectives.	1	2	3	4
3. The project had a focus on the most important problems or opportunities.	1	2	3	4
4. The project had a simple and straightforward design.	1	2	3	4
5. The project had good training programs for all involved.	1	2	3	4
6. The project had a review schedule after project completion.	1	2	3	4
7. The project had well defined and organized maintenance program.	1	2	3	4
8. The project use an appropriate Hardware and Software technologies.	1	2	3	4
9. The project recognized the level of complexity and risks.	1	2	3	4
10. The project was well-defined.	1	2	3	4
11. During the project users were aware of the requirements	1	2	3	4
12. During the project user's requirements were changed all the times.	1	2	3	4
13. Developers understood user's requirements well.	1	2	3	4

14. There was general agreement on project goals and objectives.	1	2	3	4
15. Different user departments or users had conflicting requirements.	1	2	3	4
16. The project had user’s expectations that were real.	1	2	3	4
17. During the project, the developers were familiar with the	1	2	3	4
technology base or infrastructure.	1	2	3	4
18. The project had the developed system which was very complex.	1	2	3	4
19. The projects had intensive user’s participation.	1	2	3	4
20. Senior management was highly committed to the project.	1	2	3	4
21. The project had a good project management practices.	1	2	3	4
22. Smaller Project Milestones.	1	2	3	4
23. The project had a good user-developer trust.				
24. The project had a good user-developer communication.	1	2	3	4
25. The project had a good user-developer mutual understanding.	1	2	3	4
26. The project had adequate time for development.				
27. The project had adequate developers available to work on the	1	2	3	4
project.				
28. The project had a good developer's technical expertise.	1	2	3	4
29. The project had a good developer's knowledge of the application				
domain.	1	2	3	4
30. Developer's knowledge of the systems development process was	1	2	3	4
good.				
31. The project had a good user's knowledge of the systems	1	2	3	4
development process.				
32. The project used appropriate development techniques.	1	2	3	4
33. The project used appropriate development tools e.g. (project	1	2	3	4
management				
Or CASE tools.).	1	2	3	4
34. The project applied appropriate systems development				
methodology knowledge.	1	2	3	4
35. The project had good user training.				
36. The project had a well-planned installation and adoption of the	1	2	3	4
developed system in user departments.				
37. The project had the availability of the required technology and	1	2	3	4
expertise to accomplish the specific technical action steps.				
38. The project had a good act of ‘‘selling’’ the final projects to their	1	2	3	4
ultimate intended users.				
39. The project had timely provision of comprehensive control	1	2	3	4
information at each stage in the implementation process.				
40. The project had the provision of an appropriate network and	1	2	3	4
necessary data to all key actors.				
41. The project was able to handle unexpected crises and deviations	1	2	3	4
from plan.	1	2	3	4
42. During the project, the top management was willing to provide the				
necessary resources and authority/power for project success.	1	2	3	4
43. The project had recruitment, selection and training of the	1	2	3	4
necessary personnel for the team.	1	2	3	4
44. The project covered the entire development life cycle.	1	2	3	4
45. The project had efficient coordinating and integrating of activities,	1	2	3	4
documents and personnel.				
46. The project had regular recording of any activity, mistake,				
improvement occurred during development phases.				
47. The project had sufficient testing.				
48. The project had clear and specified business priorities.				
49. The project had a clear vision and objectives				
50. Other, please specify**..**				

SECTION C: The role of system development Methodology towards IT Project success

On the lines below, place an X to indicate the type of SDM you apply in developing the system in your department:

1. To what extent is your IS department using the following standard (commercial) system development methods at present? You may mark more than one item.	To a greater extent	To some extent	To a little extent	Not at All
1.1. Structured Analysis, Design and Implementation of Information Systems (STRADIS).	1	2	3	4
1.2. Yourdon Systems Method (YSM).	1	2	3	4
1.3. Information Engineering (IE).	1	2	3	4
1.4. Rational Unified Process (RUP)	1	2	3	4
1.5. James Martin's RAD (JMRAD)	1	2	3	4
1.6. Extreme Programming (XP)	1	2	3	4
1.7. Other, please specify..	1	2	3	4

2. Which of the above standard (commercial) systems development methods were changed to fit the specific needs of your organization, and how significant were the changes? You may mark more than one item.	Did not change	Minor Change	Major Changes
2.1. Structured Analysis, Design and Implementation of Information Systems (STRADIS).	1	2	3
2.2. Yourdon Systems Method (YSM).	1	2	3
2.3. Information Engineering (IE).	1	2	3
2.4. Rational Unified Process (RUP)	1	2	3
2.5. James Martin's RAD (JMRAD)	1	2	3
2.6. Extreme Programming (XP)	1	2	3
2.7. Other, please specify..	1	2	3

3. How long has your systems development methodology been in use in your IS department?	
3.1. Less than a year	1
3.2. 1 – 2 years	2
3.3. 3 – 5 years	3
3.4. 6 – 10 years	4
3.5. 11 years or More	5
3.6. Don't know	6
4. What is the proportion of projects that are developed in your IS department by applying systems development methodology knowledge?	
4.1. None	1
4.2. 1 – 25%	2
4.3. 26 – 50 %	3

4.4. 51 – 75 % 4.5. 76 % or More	4 5

5. Please indicate the proportion of people in your IS department that apply systems development methodology knowledge	
5.1. None 5.2. 1 – 25% 5.3. 26 – 50 % 5.4. 51 – 75 % 5.5. 76 % or More	1 2 3 4 5

6. To what extent do you agree/disagree with the following statements?	Totally Agree	Agree	Neutral	Disagree	Totally Disagree
6.1. Our systems development methodology helps to develop more functional systems.	1	2	3	4	5
6.2. Our systems development methodology helps to develop more reliable systems.	1	2	3	4	5
6.3. Our systems development methodology helps to develop more maintainable systems.	1	2	3	4	5
6.4. Our systems development methodology helps to develop more portable systems.	1	2	3	4	5
6.5. Our systems development methodology helps to develop more efficient systems.	1	2	3	4	5
6.6. Our systems development methodology helps to develop more usable systems.	1	2	3	4	5
6.7. Overall, our systems development methodology helps to develop better systems.	1	2	3	4	5
6.8. Overall, our systems development methodology helps to make users more satisfied with our systems.	1	2	3	4	5

7. To what extent do you agree/disagree with the following statements?	Totally Agree	Agree	Neutral	Disagree	Totally Disagree
7.1. Our systems development methodology helps to develop new applications faster.	1	2	3	4	5
7.2. Our systems development methodology helps to improve the functionality of new applications.	1	2	3	4	5
7.3. Our systems development methodology helps to increase the productivity of the application developers.	1	2	3	4	5

7.4. Our systems development methodology helps to decrease the cost of systems development.	1	2	3	4	5
7.5. Our systems development methodology helps to improve the quality of the systems.	1	2	3	4	5
7.6. Our systems development methodology helps to decrease the cost of systems maintenance.	1	2	3	4	5
7.7. Our systems development methodology helps to improve the documentation of the systems.	1	2	3	4	5
7.8. Our systems development methodology improves the morale in my IS department.	1	2	3	4	5
7.9. Our systems development methodology helps to achieve the goals of our IS department.	1	2	3	4	5
7.10. Our systems development methodology helps to improve our IS department's reputation of excellent work.	1	2	3	4	5

8. I expect that in the next 2 years our IS department will	Totally Agree	Agree	Neutral	Disagree	Totally Disagree
8.1. Make more use of our systems development methodology.	1	2	3	4	5
8.2. Replace our systems development methodology.	1	2	3	4	5
8.3. Supplement our systems development methodology with other methodologies.	1	2	3	4	5
8.4. Abandon the use of our systems development methodology.	1	2	3	4	5
8.5. No change.	1	2	3	4	5
8.6. Other, please specify...	1	2	3	4	5

SECTION D: Systems development methods are NOT used

If your IS department does not use any systems development methods (commercial or in-house developed) as part of system development methodology, please answer the following questions.

1. Which of the following statements describe the situation in your IS department?	
1.1 Your IS department had never considered using systems development methods	1
1.2 Your IS department had considered using systems development methods, but decided against it.	2
1.2 Your IS department did use systems development methods in the past, but abandoned it.	3

2 To what extent do you agree/disagree with the following statements about the last project you were involved with?	Totally Agree	Agree	Neutral	Disagree	Totally Disagree
2.2 The profile of development projects in our IS department doesn't require the use of systems development methods.	1	2	3	4	5
2.3 Systems development methods are too complex or hard to use.	1	2	3	4	5
2.4 The current systems development practice in our IS department is adequate.	1	2	3	4	5
2.5 The experience of the developers in our IS department reduces the need for systems development methods.	1	2	3	4	5
2.6 The benefits of systems development methods use are long-term, whereas costs are incurred short term.	1	2	3	4	5
2.7 There is a lack of experienced staff in our IS department who can effectively use systems development methods.	1	2	3	4	5
2.8 New systems developed with systems development methods are not compatible with legacy systems.	1	2	3	4	5
2.9 Our IS department lacks a suitable environment to support systems development methods.	1	2	3	4	5
2.10 In our IS department there is a lack of management support for the use of systems development methods.	1	2	3	4	5
2.11 The learning curve for systems development methods is very long.	1	2	3	4	5
2.12 The financial investment in systems development methods is too large.	1	2	3	4	5
2.13 In our IS department there is a lot of uncertainty over the benefits of adopting systems development methods.	1	2	3	4	5
2.14 In our IS department there is no clear objectives for adopting systems development methods.	1	2	3	4	5

SECTION E: Project Outcome

1. Which of the following statements describe the outcome of the last system development project you were involved with?	
1.1. The project was cancelled/terminated before time.	1
1.2. The project was completed but not implemented.	2
1.3. The project was completed and implemented, but not in use anymore.	3
1.4. The project was completed and implemented, and is still in use.	4

2. If you have selected **1.4** in the previous question, for how many months has it been

If your last project were cancelled before completed, you have completed the questionnaire.
THANK YOU FOR COOPERATION!
If your last project were not cancelled before completion (i.e. you selected 2.2, 2.3 & 2.4 in question 1), please answer the following questions.

3. To what extent you agree/disagree with the following statements about the last project you were involved with?	Totally Agree	Agree	Neutral	Disagree	Totally Disagree
3.1. The project was completed on schedule.	1	2	3	4	5
3.2. The project was completed within the budget.	1	2	3	4	5
3.3. The developed system satisfied all the stated requirements.					

3.4. Speed of developing new applications was high.	1	2	3	4	5
3.5. The productivity involved in the projects was high.	1	2	3	4	5
3.6. The cost of project is low when compared to the size and complexity of the system developed.	1	2	3	4	5
3.7. The project achieved its goal.	1	2	3	4	5
3.8. Overall, the project represents excellent work.	1	2	3	4	5
3.9. Overall, the project was a success.	1	2	3	4	5

4. To what extent you agree/disagree with the following statements about the last project you were involved with?	Totally Agree	Agree	Neutral	Disagree	Totally Disagree
4.1. The functionality of the developed system is high.	1	2	3	4	5
4.2. The reliability of the developed system is high.	1	2	3	4	5
4.3. The maintainability of the developed system is high.	1	2	3	4	5
4.4. The portability of the developed system is high.	1	2	3	4	5
4.5. The efficiency of the developed system is high.	1	2	3	4	5
4.6. The usability of the developed system is high.	1	2	3	4	5
4.7. The developed system meets user's needs.	1	2	3	4	5
4.8. The documentation of the developed system is good.	1	2	3	4	5
4.9. Overall, the quality of the developed system is high.	1	2	3	4	5
4.10. Overall, the users are satisfied with the developed system.	1	2	3	4	5
4.11. Overall, the developed system is a success.	1	2	3	4	5

Any comments: ....

THANKS YOU FOR YOUR TIME AND EFFORTS IN COMPLETING THIS QUESTIONNAIRE!!!