

The relationship between process maturity models and the use and effectiveness of systems development methodologies

Christoffel Wilhelmus Janse van Rensburg

**Dissertation submitted in partial fulfilment of the requirements for the degree
Master of Science in Computer Science and Information Systems at the
Potchefstroom Campus of the North-West University**

Supervisor: Prof. H.M. Huisman

December 2012

The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NRF.

Acknowledgements

First, and foremost, I would like to thank my Heavenly Father for giving me the opportunity, guidance and support to make it to where I am today. To my parents, Piet and Salomé, Grandma Ena and Grandpa Piet; thank you for your support both financially and emotionally, without you this journey would not have been possible. To my dear friend, Stefan van Staden; thank you for the support, encouragement, and comments which often gave me quite a chuckle. To all my other friends and family members, thank you for the support and encouragement, you made this experience memorable. To my colleagues; Jacques Meeding and Waldo Brits, thank you for keeping me motivated and being there when I needed you. Lastly I want to thank my supervisor, Prof. Magda Huisman. Your role in this dissertation was critical, thank you for all your time and energy.

Ebenezer

Abstract

The need for information systems has increased to a point where virtually all business environments require some sort of software to aid in its daily operations. This study will address the need for quality information systems by examining techniques which can potentially aid in producing consistent high-quality information systems. Two techniques in particular, namely Process Maturity Models (PMMs) and Systems Development Methodologies (SDMs) are examined.

Process Maturity Models such as the Capability Maturity Model Integration (CMMI) as well as the ISO-9000 standards aid in standardising and improving an organisation's information systems development processes. These Process Maturity Models often require either the use of certain Systems Development Methodologies or at the very least techniques used within some Systems Development Methodologies. Systems Development Methodologies refer to a set of development processes, tools, techniques etc. which can be used during software development to standardise the entire development process by offering the use of modelling techniques, tools to analyse requirements, illustration of processes etc. These techniques differ from one Systems Development Methodology to the next.

This study aims to identify the relationship between Process Maturity Models and Systems Development Methodologies. During the research process a questionnaire was sent out to people within the information technology business environment. The questionnaire contained questions used to determine and measure the usage of Systems Development Methodologies and how projects were affected. The questionnaire was also used to do an informal assessment of each respondent's Capability Maturity Model level. Furthermore the data retrieved was statistically analysed and the results were interpreted.

The results indicate that a relationship exists between the use of SDMs and the success of the respondent's development processes and developed products. A total of 73% of respondents indicated that they do use SDMs to some extent, the most common being the Systems Development Lifecycle (SDLC). The majority of organizations implementing SDMs have been doing so for three years or more. Results also show that most of the respondents are not certified in some formal Process Maturity Model; however, they do implement some

of the processes required by models such as the CMMI. An informal assessment performed indicated that 65% of respondents can be grouped into a perceived CMMI level 2 category. Project outcome was measured and the relationship between PMM implementation as well as SDM use was measured. Results show no statistical evidence which indicates that an organisation's perceived CMMI level is influenced by SDM use, both vertically and horizontally. Results do, however, indicate that organizations which have been implementing SDMs for a longer period of time are more likely to apply CMMI level 4 activities. Results also indicate that the horizontal use (number of projects/people which implement SDM knowledge) of SDMs have a significant effect on the development process- and the developed product success. Lastly the results indicated that organizations which satisfy more of the CMMI's level 4 activities experience a higher quality development process which leads to a more successful development process.

Keywords

Systems Development Methodologies, Process Maturity Models, Information Systems Development, Capability Maturity Model Integration, CMMI, Software Process Improvement, SPI, ISO 9000, Scientific-Positivism, Questionnaire, Survey.

Abstrak

Die behoefte aan inligting stelsels het gegroei tot op 'n punt waar werklik elke tipe werksomgewing een of ander vorm van 'n gerekenariseerde sagteware stelsel benodig in hulle daaglikse werkverrigting. Hierdie studie spreek die behoefte van hoë-gehalte inligting stelsels aan deur tegnieke wat potensiaal het om te kan bydra tot die ontwikkeling van konstante hoë-gehalte inligting stelsels. Twee tegnieke in besonder word bestudeer, naamlik Proses-volwassenheidmodelle (*Process Maturity Models - PMMs*) asook Stelselontwikkelingsmetodologieë (*Systems Development Methodologies - SDMs*).

Proses-volwassenheidmodelle soos die *Capability Maturity Model Integration* (CMMI) sowel as die ISO-9000 standaard help om 'n organisasie se inligtingstelselontwikkelingsprosesse te standardiseer. Proses-volwassenheidmodelle benodig gereeld die gebruik van Stelselontwikkelingsmetodologieë of ten minste van die tegnieke wat gebruik word binne Stelselontwikkelingsmetodologieë. Met Stelselontwikkelingsmetodologieë word daar verwys na 'n stel ontwikkelingsprosesse, gereedskap, tegnieke, ens. wat gebruik word gedurende sagteware ontwikkeling om sodoende die volledige ontwikkelingsproses te standardiseer. Tegnieke soos modelleringsdiagramme, gereedskap om vereistes te analiseer, grafiese illustrasies om prosesse te demonstreer ens. word gebruik gedurende die ontwikkelingsproses. Hierdie tegnieke verskil van een Stelselontwikkelingsmetodologie na 'n ander.

Hierdie navorsing fokus daarop om die verhouding(s) tussen Proses-volwassenheidsmodelle en Stelselontwikkelingsmetodologieë te identifiseer. Gedurende die navorsing is gebruik gemaak van vraelyste wat uitgestuur is na persone binne die inligtingstechnologie besigheidsomgewing. Die vraelyste het vrae bevat wat gebruik is om te bepaal en te meet hoe intensief Stelselontwikkelingsmetodologieë toegepas word gedurende die ontwikkelingsproses en hoe dit projekte geïmpakkeer het. Die vraelyste is ook gebruik om 'n informele volwassenheidsvlak vir elke respondente te bepaal. Verder is die resultate statisties ontleed en geïnterpreteer.

Resultate toon aan dat daar 'n verwantskap is tussen die gebruik van Stelselontwikkelingsmetodologieë en die sukses van die respondente se

ontwikkelingsprosesse asook die eindproduk. 'n Totaal van 73% van die respondente het aangedui dat hulle SDMs toepas op een of ander wyse, en die meerderheid hiervan gebruik die *Systems Development Lifecycle* (SDLC). Dié organisasies wat SDMs gebruik gebruik dit al vir drie jaar of meer. Die resultate toon ook dat die meerderheid van respondente se organisasies nie gesertifiseer is in een of ander formele prosesmodel nie; 'n informele assessering het wel aangedui dat 65% van respondente in 'n waargenome CMMI vlak 2 val. Projekresultate is gemeet en die verwantskap tussen PMM implementasie asook SDM gebruik is bepaal. Resultate toon aan dat daar geen statistiese verhouding bestaan tussen die waargenome CMMI vlak van 'n organisasie en hulle SDM gebruik nie (beide horisontale en vertikale gebruik). Resultate toon aan dat organisasies wat wel SDMs vir 'n langer tydperk gebruik meer waarskynlik is om alreeds van CMMI se vlak 4 aktiwiteite toe te pas. Die resultate toon ook aan dat die horisontale gebruik (die aantal projekte/persone wat SDM kennis toepas) van SDMs 'n merkwaardige invloed het op die ontwikkelingsproses asook die ontwikkelde produksukses. Laastens dui resultate ook daarop dat organisasies wat aan meer van CMMI se vlak 4 aktiwiteite voldoen 'n hoër kwaliteit ontwikkelingsproses ervaar wat lei tot 'n meer suksesvolle ontwikkelingsproses.

Sleutelwoorde

Stelsels-ontwikkelingmetodologieë, Proseses-volwassenheidsmodelle, Inligting
Stelselsontwikkeling, *Capability Maturity Model Integration*, CMMI, Sagteware
Prosesverbetering, SPI, ISO 9000, Wetenskaplike-Positivisme, Vraelys.

Contents

Acknowledgements.....	ii
Abstract.....	iii
Keywords.....	iv
Abstrak.....	v
Sleutelwoorden.....	vi
Contents	vii
Chapter 1 - Problem Statement.....	1
1.1 Introduction	1
1.2 Systems Development Methodologies and Process Maturity Models.....	1
1.3 Research aims and objectives	5
1.4 Method of investigation.....	7
1.5 Outline of the dissertation	8
1.6 Chapter summary.....	9
Chapter 2- Systems Development Methodologies and Process Maturity Models.....	10
2.1 Introduction	10
2.2 Systems Development Methodologies	11
2.2.1 Historical origin and use of Systems Development Methodologies	13
2.2.2 Advantages of using systems development methodologies.....	16
2.2.3 Disadvantages or criticism of systems development methodologies.....	17
2.3 Types of systems development methodologies	18
2.3.1 Process-oriented systems development methodologies.....	20
2.3.2 Data-oriented systems development methodologies	24
2.3.3 Object-oriented systems development methodologies	27
2.3.4 Rapid systems development methodologies	30
2.3.5 People-oriented systems development methodologies.....	32
2.3.6 Organizational-oriented systems development methodologies	34
2.4 Systems development methodology comparisons.....	36
2.4.1 Methodology comparisons	37
2.4.2 Methodology comparisons conclusion	41
2.5 Introduction to PMMs.....	41
2.6 ISO 9000	43
2.6.1 The origin and history of ISO 9000.....	43
2.6.2 ISO 9000 Certification requirements	44

2.6.3	Advantages of being certified in ISO 9000	48
2.6.4	Criticisms against the implementation of ISO 9000.....	49
2.7	Capability Maturity Model Integration (CMMI).....	49
2.7.1	The origins and historical use of CMMI	50
2.7.2	Understanding CMMI levels an overview	51
2.7.3	Advantages of being certified in CMMI	57
2.7.4	Criticisms against the implementation of CMMI	58
2.8	CMMI and Systems Development Methodologies integration	59
2.9	Previous research on Process Maturity Models and Software Development Methodologies integration	60
2.10	Chapter summary.....	64
Chapter 3 - Research Design		65
3.1	Introduction	65
3.2	Research Paradigms.....	66
3.2.1	Scientific-Positivism	67
3.2.2	Interpretive	67
3.2.3	Critical social	68
3.3	Research paradigm used in this dissertation	68
3.3.1	The Scientific Research Paradigm	69
3.3.2	The Scientific method	69
3.3.3	Criticisms of the positivist research paradigm	72
3.3.4	Research methods used in the positivistic paradigm	72
3.3.5	Survey data gathering methods.....	75
3.3.6	Planning and designing a survey.....	75
3.3.7	Response rate and non-responses:.....	77
3.3.8	Sample size:.....	78
3.3.9	Data generation methods	78
3.3.10	Data-generation method used in this research:	78
3.3.11	Application of data-generation method	79
3.3.12	Data analysis	81
3.4	Chapter Summary	82
Chapter 4 – Research Results		83
4.1	Introduction	83
4.2	Development Environment.....	83

4.2.1	Business area	83
4.2.2	Role within the organization.....	85
4.2.3	Organization's years in operation.....	85
4.2.4	Size of the organization.....	86
4.2.5	Size of the ISD department	87
4.2.6	Information System Development (ISD) Skill Level.....	88
4.2.7	Software Procurement.....	89
4.2.8	Project outcome.....	90
4.2.9	Project size	91
4.2.10	Development Environment Summary.....	92
4.3	SDM Usage and Process Maturity certification	92
4.3.1	SDM usage.....	93
4.3.2	Reasons for not using an SDM	93
4.3.3	Historical SDM usage	95
4.3.4	Horizontal methodology use.....	96
4.3.5	Vertical methodology use	99
4.4	Process Maturity Model certification	101
4.4.1	Perceived CMMI Level.....	104
4.5	Relationships between PMM certification and SDM usage success factors	108
4.5.1	Information Systems Development Success.....	109
4.5.2	Factors which influence the relationship between the IS development process- and the IS developed product success.....	113
4.6	Relationships of the Horizontal- and Vertical SDM Usage, SDM Usage Strictness, and the SDMs Time in Use in regards to the perceived CMMI Levels	115
4.7	Chapter Summary	116
Chapter 5 – Summary and Final Research Conclusions		117
5.1	Introduction	117
5.2	Research Results and contributions.....	117
5.3	Limitations.....	119
5.4	Conclusions	120
5.5	Future research.....	120
5.6	Closing.....	120
Appendix A – Research Questionnaire		122
References		130

Chapter 1 - Problem Statement

1.1 Introduction

Throughout the Information Systems Development timeframe, a certain problem has become clear: Many difficulties are encountered when developing a successful Information System. The term *successful* refers to staying within budget, completed on or before schedule and completely meeting the client's expectations of the system's look-and-feel, capabilities, responsiveness, etc. In order to alleviate these issues we are focusing on Process Maturity Models (PMMs) as well as the use of Systems Development Methodologies (SDMs). It should also be noted that the above-mentioned solutions are merely two of a host of possible approaches which assist in Information Systems Development.

Many companies claim that they use Systems Development Methodologies in developing their software systems (Hansen & Kautz, 2005:6; Iivari & Maasaari, 1998:501; Rahim *et al.*, 1998:949). This study focuses on Systems Development Methodologies and their relationship with Process Maturity Models such as the Capability Maturity Model Integration (CMMI) and ISO90003. In order to be certified in either CMMI or ISO the company in question needs to use Systems Development Methodologies in its project development and maintenance processes (Curtis, 1992; Paulk *et al.*, 1993; Huisman & Iivari, 2000).

This study will examine the relationship between Process Maturity Models and the use and effectiveness of systems development methodologies.

In this chapter Systems Development Methodologies and Process Maturity Models are briefly discussed. The research aims and objectives are briefly discussed as well as the method of investigation used to collect the required data.

1.2 Systems Development Methodologies and Process Maturity Models

Information Systems Development has seen huge growth during the last decade and although it has been in existence for at least 40 years, developers are still experiencing

difficulties in delivering successful systems. According to the 2009 CHAOS reports (The Standish Group, 2009) as little as 32% of IT projects developed during 2009 could be considered as succeeding (projects delivered on time, on budget and with required features and functions). 44% were challenged (projects were late, over budget, and/or with less than the required features and functions); the remaining 24% failed (projects cancelled prior to completion or delivered and never used). These figures are even poorer than in previous years. There are, however, studies that contradict the finding of the CHAOS reports, such as the study done by Eveleens and Verhoef (2010) as well as Jørgensen & Moløkken-Østvold (2006) which indicated that if using own data (instead of using the Standish group's data) they found that the Standish definitions of *successful* and *challenged* projects have four major problems: they're misleading, one-sided, they distort the estimation practice, and result in meaningless figures. This is in reference to the Standish group's (The Standish Group, 2009) definitions for a successful project. In the CHAOS reports project success is measured according to the following factors:

- Project completed on time
- Project completed within budget
- Project offers all the functions as initially specified.

Eveleens and Verhoef (2010) indicate that merely categorizing projects according to these factors can lead to inaccurate results. They continue by stating that the CHAOS reports incorrectly categorize a successful project as challenged when for example, the project is completed on time and within budget but lacks a miniscule amount of the original planned functionality. Eveleens and Verhoef (2010) also stated that the measurements used in the CHAOS reports are singly focused on estimation deviation.

In spite of the latter criticism, the CHAOS reports are often quoted in the System Development environment (Egorova *et al.*, 2010; Garousi & Varma, 2010; Jørgensen, 2011). This indicates that, globally, information software development is still experiencing a crisis. Various examples of failed systems exist in the popular media, a few examples include: The highly publicised project called the Virtual Case File (VCF) which cost the FBI \$170 million was called off in 2005 prior to its completion which in turn resulted in various congressional hearings and over \$100 million in overruns (Na *et al.*, 2007). Another example famous for its

estimated cost of around \$400 million was the 'Everest' system designed by the Ford Motor Company along with Oracle. Due to the project's sheer size, the amount of work and process flow was simply too complex and resulted in unsatisfied users, missing functionalities, and a large number of other problems which led to the decision being made to abandon the project (Keefe, 2004).

Furthermore studies by McManus and Wood-Harper (2010) have shown that project failures across the European Union resulted in an estimated €142 billion loss in investments during 2004. A study by Jiang *et al.* (1998) focuses on the perception of problems/failures by examining the different stakeholder groups and what they perceive as problems at different levels of occurrence.

In this research two possible solutions to the said software crisis are examined. The first solution, Systems Development Methodologies, offers techniques which have been tried and tested and proven to aid in developing successful information systems; and the second, Process Maturity Models, aids in standardising an organisation's development processes and quality management. Both of these techniques are discussed in further detail throughout the research.

A Systems Development Methodology can be defined as a collection of tools and techniques used during the development, or part of the development of information systems. These development techniques are normally based on a set of rationales and an underlying philosophy recommendation within a specific context. Examples of techniques used in information systems development include phases, procedures, tasks, rules, techniques, guidelines, documentation and tools. It is also possible for systems development methodologies to recommend certain management procedures whilst also identifying potential participants along with the needed training requirements (Avison & Fitzgerald, 2006: 568).

Another definition for a Systems Development Methodology as discussed by Huisman and livari (2006:32), where they break down Systems Development Methodologies into four main parts, is as follows:

- Systems Development Approach: The philosophical foundation on which the methodology is established.
- Systems Development Process Model: The representation of cycles or stages through which a system is built.
- Systems Development Method: A systematic approach of developing the system, consisting of the required documentation, activities, tools to be used, etc.
- Systems Development Technique: Procedures followed during development activities.

The Systems Development Approach can be used to classify a few available Systems Development Methodologies. This is discussed in further detail in Chapter 2. The advantages offered by implementing SDMs provide an opportunity to alleviate common issues experienced during information software development. Some of these advantages include: improving the development process, delivering a better end-product and standardising the development processes and procedures. These and other advantages will be discussed further in 2.2.2.

One of the questions this research aims to address is whether the use of Systems Development Methodologies, a requirement in Process Maturity Model Certification, benefits the organization. The secondary objective of the research is to ascertain whether the pressure to get a CMMI certification, and thus the need to implement a Systems Development Methodology, forces organizations to implement processes which they are not accustomed to or they do not have the required skill sets to successfully implement (Fitzgerald, 1998:317).

Research done by Huisman and Iivari (2000) on the subject of the perceived maturity of IS departments and the deployment of systems development methodologies has shown that organizations roughly follow the maturity levels as stated in CMM. The study also showed that as an organization's maturity increases, so does its usage of systems development methods.

Other research, such as that done by van der Pligt *et al.* (1997:273), has shown that quality certificates can give a wrong impression in regards to the organization's real development

capabilities. They also continue by stating that this, however, should not lead to the abolition of standards.

Most Process Maturity Models are based on the Software Engineering Institute's Capability Maturity Model. These practices measure an organization's development processes against a set of standards compiled by SEI (Carnegie Mellon Software Engineering Institute, 2009; Daymond, 1995).

Organizations can increase their CMMI-level by complying with more maturity tasks. By attaining a CMMI level certification and by raising their maturity level, organizations are likely to experience the following advantages (Staples & Niazi, 2007):

- Raise the organization's project development performance
- Improve the quality of the developed product and process management
- Increase possible clients' perceived image of the organization
- Increase the organization's competitive standing.

Organizations which implement and maintain strategic assets, such as CMMI or ISO certification, are reported to experience long-term competitive advantages which, in turn, also ensure the firm's cost advantages (Markides & Williamson, 1994). The importance of this study lies in the research done on whether the use of Systems Development Methodologies in these Process Maturity Models will actually benefit an organization. Many organizations invest large amounts of money in order to adhere to standards set by these models and consequently suffer great losses should these implementations fail.

1.3 Research aims and objectives

The main aim of this research is to study whether a relationship exists between Process Maturity Models and the use and effectiveness of systems development methodologies.

Objectives that this research will address can be listed as follow:

1. Attain background information on organisations' usage of systems development methodologies during the development of information systems.
2. Determine each organisation's process maturity level.
3. Determine each organisation's level of Process Maturity Model usage.
4. Determine the relationship (if any) between the Process Maturity level of the organisation and the use and effectiveness of Systems Development Methodologies.
5. Determine the relationship between an organisation's development process success and its use of Systems Development Methodologies as well as the organisation's perceived CMMI level.
6. Determine the relationship between an organisation's developed product success and its use of Systems Development Methodologies as well as the organisation's perceived CMMI level.

The following figure illustrates these objectives:

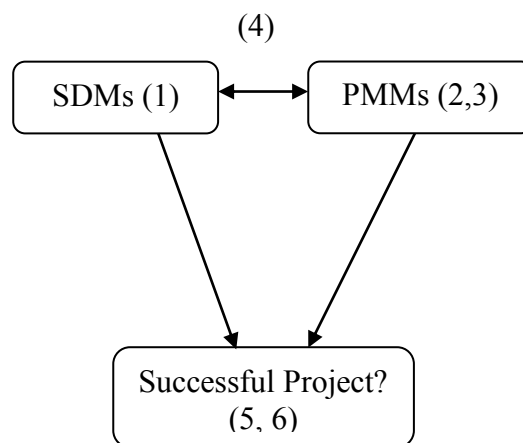


Figure 1.1 – Research Objectives

1.4 Method of investigation

This research is based on the scientific positivism research approach. Three main paradigms exist, namely the positivist approach, the interpretive approach, and the critical social approach (Kuhn, 1970; Kuhn, 1996; Lukka, 2010). One of the reasons why the positivist research approach is used, as opposed to one of the alternatives, can be stated as follows:

The positivist perspective focuses primarily on scientifically verifiable results. In other words, by using questionnaires and appointing numerical values to each question's possible answer(s) statistical analysis can be used on the results to compute numerical values. These values can be used to indicate certain trends or similarities within the data (Crombie, 1996; Winther, 2012).

Possible positivist research methods include:

- Surveys,
- Case studies, and
- Experiments.

A survey was used as the chosen research method, due to the nature of the data at hand. Data was collected from a large number of participants in a standardized manner. Two hundred companies were contacted to participate in the survey of which eighty responded. The data was then analysed using statistical methods in order to determine patterns or similarities within the data. According to Oates (2006) surveys are the most commonly used research method and they are also typically associated with the positivist research paradigm which is the paradigm on which this research was postulated.

For this research questionnaires were used as a data generation method. The questionnaire used in this research (see appendix A) contained a total of twenty-two question divisions, each division linking to the research objectives. Questionnaires offer the opportunity to establish a list of pre-defined questions, listed in a particular order. Generally questionnaires offer answers in the form of multiple choice, and this helps to standardize answers and thus offer the opportunity to analyse them for statistical relevance and relationships within the data. The questionnaire used in this research mainly contained

multiple choice answers with the odd open-ended question to provide respondents with the opportunity to indicate which Systems Development Methodology(-ies) they use in their development process.

Lastly the data was analysed. The data can be analysed in either qualitative or quantitative form. The data that was generated in this research was quantitative data. Techniques used to summarize the data varied from factor analysis, regression, means, cluster analysis, and correlation matrices.

1.5 Outline of the dissertation

Chapter 1 - Problem statement:

In this chapter it is stated that there is a software crisis and that this research focuses on Process Maturity Models and Systems Development Methodologies to alleviate these crises. Furthermore the latter concepts are briefly discussed; the research aims and objectives are provided, the research method used is briefly discussed, and a chapter division is provided.

Chapter 2 - Systems Development Methodologies and Process Maturity Models:

In chapter 2 Systems Development Methodologies and Process Maturity Models such as CMMI and ISO 90003 were examined. The techniques required by CMMI were summarized and theoretically compared to ascertain which of these techniques are used in a few common Systems Development Methodologies.

Chapter 3 - Research Design:

The different research paradigms are briefly discussed. The chosen paradigm (scientific positivism) approach is examined; criticisms, advantages, disadvantages, as well as the data collection method (questionnaires) and data analysis are discussed.

Chapter 4 - Research results:

Research results are provided; descriptive statistics summarize the organization's size, area of expertise, skill levels, etc. Furthermore the relationship between the usage of Systems Development Methodologies and Process Maturity Models is examined and summarized.

Chapter 5 - Conclusions, interpretation, discussion and recommendations:

In this chapter the research results are examined, conclusions drawn and discussed, and recommendations concerning the topics are made.

1.6 Chapter summary

In this chapter the problem statement was identified and the need for some kind of solution to ensure successful Information Systems Development was provided. Two of these solutions are briefly reported (Systems Development Methodologies and Process Maturity Models), the method of investigation is briefly discussed, and chapter divisions are provided. The next chapter, Chapter 2, focuses on Systems Development Methodologies and Process Maturity Models (Capability Maturity Model Integration and ISO 90003).

Chapter 2- Systems Development Methodologies and Process Maturity Models

2.1 Introduction

This chapter focuses on the different types of systems development methodologies. The term (systems development methodologies) is defined and its historical use examined. Advantages and critique against the use of systems development are reviewed. A list of the most commonly associated systems development methodologies within each category is specified. One systems development methodology from each group is discussed in more detail in order to form a better understanding of that type. These systems development methodologies are compared in a grid in order to further summarise and compare each.

This is followed by an examination of Process Maturity Models; Process Maturity Models are defined, different types of Process Maturity Models are discussed; ISO 9000-3 and the Capability Maturity Model Integration are focused on in particular. The previously mentioned Process Maturity Models are examined focusing on their historical development and use, the advantages of implementing Process Maturity Models, as well as the critique against the use of these Process Maturity Models. The chosen Systems Development Methodologies are compared to the techniques required by the Capability Maturity Model Integration, and finally previous research on the subject is reviewed.

Figure 2.1 provides a basic framework of this chapter:

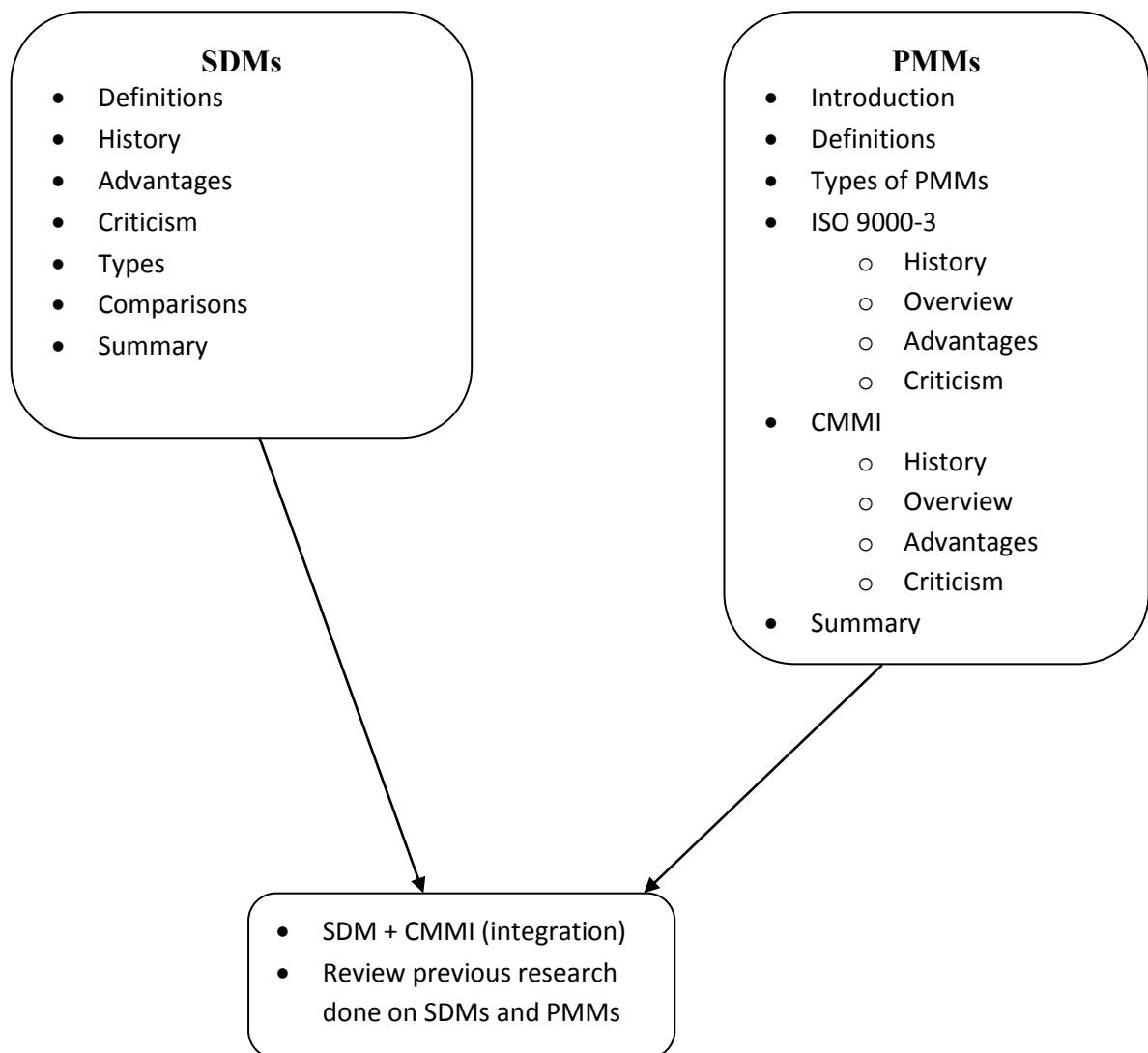


Figure 2.1 - Chapter Framework

2.2 Systems Development Methodologies

No accurate or exact definition of a Systems (Software) Development Methodology exists. Some argue that the term “methodology” has no place in Information Systems as the term is literally defined as a “science of methods” Baskerville (1992). Avison and Fitzgerald (2006:567) argue that the term methodology encapsulates more concepts than a method. Therefore a methodology contains characteristics that are not implied by method, it includes a philosophical view.

Avison and Fitzgerald (2006:567) define Systems Development Methodologies as a collection of tools and techniques used during the development, or part of the development, of information systems based on a set of rationales and an underlying philosophy recommendation within a particular context. This includes phases, procedures, tasks, rules, techniques, guidelines, documentation and tools. The methodology can also include recommendations concerning the management and organization of the approach and the identification and training of the participants.

Huisman and livari (2006:32) define a Systems Development Methodology by examining whether it can be broken down into four main parts, which are:

- A systems development approach: The philosophical foundation on which the Systems Development Methodology is built, for example the total development time needs to be minimised and development should cater for changes in scope (agile development).
- A systems development method: The tools, documentation, activities, etc. used to develop the system for example firstly the requirements should be gathered, afterwards the durations and schedule should be determined and so forth.
- A systems development process model: The representation of steps that are followed to build the system for example sequential – Each step is followed by a logical successor.
- A systems development technique: The procedures that are followed and used to aid in developing a system for example Unified Modelling Language (UML).

These four concepts are also used later in this chapter to compare some of the most commonly used Systems Development Methodologies.

Furthermore Chan and Thong (2009:803) define Systems Development Methodologies as “a documented collection of policies, processes, and procedures” which aids in increasing the overall quality of the product (system being developed) and improve the developing party’s development productivity.

Lastly, by adopting and adding to the above-mentioned definitions, a methodology can be defined as follows:

A Systems Development Methodology can be defined as a collection of development processes, tools and techniques, documentation standards, policies, and procedures which are used to develop a part of, or an entire Information Technology (IT) project. A methodology consists of four core areas, namely:

- Philosophical Systems Development Approach: What is the main aim (referred to as the philosophical approach) of this methodology? (Speed, agility, stability, etc.) This provides the groundwork for the rest of the methodology, as it implies which of the objectives will have priority;
- Systems Development Process Model: To develop a successful Information Technology System, the development process needs a process model which represents the development cycles and/or stages through which the system is built;
- Systems Development Methods: The steps to be followed to develop the system; consisting of the required documentation, tools, techniques, activities, etc. and;
- Systems Development Technique: The aids which are used during the development to help execute activities.

All these areas aid in standardising Systems Development in order to raise product quality as well as increase development efficiency and reduce development time, etc.

2.2.1 Historical origin and use of Systems Development Methodologies

In the early days of information systems, the use of detailed development processes and procedures was not yet required. The need for Systems Development Methodologies only started to arise when systems had to address multiple issues and their complexity increased. In this section we examine the origins of Software Development Methodologies and review how the development and implementation of information systems evolved over time.

In a study by Avison and Fitzgerald (2006:28) as well as their book (2006:576-589) the authors identify and discuss four eras of information systems development; namely: The

pre-methodology era, the early methodology era, the methodology era, and lastly the post-methodology era. In this study those four eras will also be used to classify the different eras through which systems development methodologies evolved.

1. The pre-methodology era (early 1970s)

In this era (known as the pre-methodology era) information systems were still being developed without implementing any form of standardised development methods. During this period the main focus of information systems was on solving various challenging technical and mathematical problems by means of programming. The hardware available during these early days was one of the main factors that determined the limitations of the required software. This resulted in developers who were technically trained to solve the relevant problems; however, they never really fully understood the business aspects in which the systems were to be deployed. The difference in highly technical knowledge and the organisational needs of the business resulted in various communication issues that limited the development potential. This, along with other factors, resulted in systems which would normally be too difficult to fully accommodate the end-users. Despite these problems, however, the demand for additional information systems quickly increased as its potential became evident.

2. Early methodology era (1970s to early 1980s)

In this era the apparent need for some form of development standards became clear. Computer-based applications started to focus on the identification of processes and phases involved in development which would aid in establishing disciplines to manage these development processes. This was the start of the “waterfall” approach which would later be known as the Systems Development Life Cycle (SDLC). Various versions of the SDLC existed, most however contained the following main stages: feasibility study, systems investigation, analysis, design, development, implementation and management. These stages followed each other sequentially.

3. Methodology era (mid 1980s to the late 1990's)

Due to various criticisms against the SDLC, a host of other systems development methodologies began to emerge. This led to what was referred to as the 'era of methodologies'. Seeing that so many systems development methodologies began emerging the term 'methodology' was used to describe these different approaches. These new systems development methodologies could mostly be traced back to one of two sources, namely developed from practice, or developed from theories. All, if not most, of these methodologies relied on a key concept or similar concepts; this of course refers to data flow diagrams or entity relationship diagrams, but usually not both. Over time these processes formed into what became known as methodologies. As time progressed these methodologies continued to evolve and spawn variants, as after each implementation the methodology would be reviewed and edited depending on the results achieved. Eventually this led to methodologies which became well documented; stayed consistent; updated when needed; researched and further developed; marketed; evolved into training packages; and later on provided along with supporting software.

4. Post-Methodology era (late 1990s to now)

The current era is identified as the post-methodology era; this refers to the sense that methodologies are perceived as having evolved beyond the pure methodology era. Reflection occurs on the usage of said methodologies; the beneficial aspects of systems development methodologies are examined and reconsidered. In the past systems development methodologies were often seen as a universal remedy for difficulties experienced in information systems development. This resulted in systems development methodologies being implemented simply due to the fact that the adoptees felt that they should implement a process to aid in managing their development processes, along with unrealistic expectations of their application. As a result various projects failed to deliver on expectations and alternative solutions were sought; in some cases these alternatives aided in delivering consistent successful projects, in other cases the results were even worse than before. All of these factors lead to some people rejecting systems development methodologies purely on the principle of avoiding it at all costs. This however does not indicate that systems development methodologies are not successful; merely that as

requirements continue to evolve and migrate, problems or unexpected shortcomings will occur and systems development methodologies will continue to adapt and cater for the said problems.

2.2.2 Advantages of using systems development methodologies

Systems Development Methodologies offer various advantages, ranging from an increase in information system quality to a more efficient development environment. As a summary the following table can be reviewed:

Table 2.1 – Advantages of using methodologies

Advantages	Source
Systems Development Methodologies aides in standardizing the development process. SDMs can cover physical design, conceptual issues, procedures or the whole range of intermediate stages. During the development process a standardized process is followed which aides in integrating systems.	Fitzgerald and Avison (2006: 572) Futrel <i>et al.</i> (2002:107) Yahya <i>et al.</i> (2002:15)
A Systems Development Methodology offers essential supporting tools which aid in completing difficult practices. It can range from being designed to solve particular types of problems to an all-encompassing general purpose Systems Development Methodology.	Yahya <i>et al.</i> (2002:15) Saini <i>et al.</i> (2009:89)
Aides in developing a better system. A methodology may, or may not, include tools and toolsets such as charting software, CASE tools, word processing etc. which ease project development and illustrate key concepts to stakeholders, users, etc. These tools have also been tried and tested; proven to be useful.	Fitzgerald & Avison (2006: 24, 570) Yahya <i>et al.</i> (2002:26)
SDMs attempt to aid in developing a better end product. This includes finishing a project within schedule and budget while meeting user requirements as well as expectations.	Fitzgerald & Avison (2006: 24, 570) Huisman & livari (2006:33) Yahya <i>et al.</i> (2002:15) Masrek <i>et al.</i> (2008:143)

By examining these advantages we can identify why systems development methodologies offer viable solutions to the software development crisis. Systems can be developed using standardised techniques which have been proven to aid in delivering consistent quality systems.

2.2.3 Disadvantages or criticism of systems development methodologies

Criticism or disadvantages regarding Systems Development Methodologies exist; according to Avison and Fitzgerald (2006: 38-43) the following weaknesses can be considered:

- Unstable. Business processes change, requirements change, etc. and this results in unstable development processes.
- Failure to meet the exact needs of management. Potential shortcomings may occur as a 'fix-it-all' solution is unlikely to completely cover all of the required aspects.
- Complexity. Processes are often more complex than initially anticipated and this results in the development process becoming inherently more complicated.
- Inflexibility. Outputs are usually determined in the early stages of the design process and thus output requirements change closer to the project completion, the adjustment becomes increasingly difficult to implement.
- Workload. Functionalities are designed without spending the proper time on identifying all the requirements.
- Unsatisfied users. Users often reject the system at hand due to complexity of their inability to correctly communicate their expectations.
- Documentation. When documentation is overly technical the average user won't be able to fully understand its contents.
- Backlog. Scheduling may cause a backlog to occur. This results in systems development being postponed until enough time becomes available to complete the required work
- Impact on other systems. Often the effect of the required system on existing systems is ignored which may lead to conflicts.

Further criticisms from various other authors can also be summarized as follows:

- Structured methods are primarily generic; they aim to be applicable to as wide as possible a range of projects, and this results in their not being able to fully address certain types of businesses' needs (Hardy *et al.* 1995:467).
- Individual structured methods are tailored to cater for a specific range of systems development projects, which can only effectively be applied to those areas (Gillies & Smith. 1994).
- Structured methods developers are concerned with developing a methodology that is internally consistent and they rigorously describe its stages/steps which need to be followed in order to effectively develop a system. This results in a wide array of techniques which can result in confusion when being applied in the development of particular projects (Olle *et al.*, 1991).

When reviewing the criticisms to systems development methodologies one might feel disheartened about implementing said techniques; however, when comparing the criticisms to the advantages on offer, the potential benefits outweigh the potential drawbacks. Thus we feel that systems development methodologies offer a logical and tangible solution to develop information systems.

2.3 Types of systems development methodologies

Systems Development Methodologies can be categorised in various ways. In this section we examine possible categorisation techniques; the chosen technique is described in further detail, and the said technique is applied to identify an example Systems Development Methodology within each category.

According to Fitzgerald and Avison (2006:568) systems development methodologies can be categorised by examining each systems development methodology's philosophical approach. This approach is used in this research; each category is briefly summarised by stating its core philosophy which is then followed by a detailed review and explanation of each category. This leaves the following possible classifications, along with their respective philosophical approaches:

- Process-oriented systems development methodologies (reviewed in 2.3.1). Process-oriented systems development methodologies focus on the processes involved in the system.
 - Functional decomposition. The system is decomposed into its core functions.
 - Process modelling. The processes involved/required in the system are identified and illustrated.
- Data-oriented systems development methodologies (reviewed in 2.3.2). Development is data-driven. Processes may change but data stays consistent.
 - Strategic information systems. The system aims to strategically address the required functionalities in order to meet requirements.
 - Data modelling. There is a focus on designing the system around the data which will be captured and used within the system.
- Object-oriented systems development methodologies (reviewed in 2.3.3). These systems development methodologies focus on identifying re-usable object and classes.
 - Object-orientation. Processes are broken down into re-usable objects and classes.
- Rapid systems development methodologies (reviewed in 2.3.4). These systems development methodologies focus on short term development, offering as many of the functionalities in as brief as possible a timeframe.
 - Rapid application development. Applications are developed in quick development “sprints”.
 - Agile systems approach. Short development periods which can handle changes in project scope/requirements.
 - Object-orientation. Re-usability in terms of delivering similar functions in different projects/processes.
- People-oriented systems development methodologies (reviewed in 2.3.5) focus on all the people involved in and affected by the development process.
 - People-oriented. Focuses on the people affected by the system as well as addressing their requirements.

- Social-technical. Focuses on the social aspects of the system. How the system will affect society.
- Organizational-oriented systems development methodologies (reviewed in 2.3.6) focus on the organisation as a whole.
 - Systems theory.

This approach serves as this research's method of categorising Systems Development Methodologies.

Other frameworks used to classify Systems Development Methodologies exist, one such example is the framework suggested by Iivari *et al.* (2001:186-189). According to the authors, the Information Systems Development (ISD) Paradigm, as well as the ISD Approaches, can be used to uniquely categorise Systems Development Methodologies. Although the author's categorisation is unique, similarities can be drawn between the latter framework and the one suggested by Avison and Fitzgerald (2006:568). For example, both frameworks emphasise the fact that a (philosophical) approach can be used to identify the systems development methodology's main focus.

2.3.1 Process-oriented systems development methodologies

Process-oriented systems development methodologies are the mechanisms and skills used to effectively and efficiently implement planning, management and policy activities that involve a number of people interacting, and are often used in decision-making. Process-oriented methodologies provide structured approaches in order to reach the desired outcomes. An emphasis on identifying processes exists; using techniques such as functional decomposition, decision trees, decision tables and data flow diagrams to aid in identifying and illustrating processes (Floyd, 1981; Howard *et al.*, 1999).

Examples of process-oriented methodologies include:

- Structured analysis, design, and implementation of information systems (STRADIS) – Gane and Sarson (1979).
- Yourdon System Method (YSM) – Yourdon (1993).
- Jackson Systems Development (JSD) – Jackson (1983).

To further understand process-oriented methodologies the following summary of STRADIS can be reviewed:

Structured Analysis, Design, and Implementation of Information Systems (STRADIS)

Development Method of STRADIS

Structured analysis, design, and implementation of information systems

1.1 Initial study

With the initial implementation of the systems development methodology, an attempt is made to ensure that the chosen systems will be those which will be warranted in the competitive environment before even starting development. It is generally accepted that the most important criterion for this process is monetary costs and benefits of each proposal. System analysts conduct the initial study, gathering data from relevant users and managers, all in an attempt to assess the proposal with regard to its effect on systems development within the organisation. During this process, which normally lasts between two days and four weeks, the existing system and its interfaces should be illustrated using an overview dataflow diagram; the schedule as well as the costs involved should also be indicated.

During the initial study a report should be generated. The report must then be reviewed by the relevant management as the decision whether to conduct a more detailed study rests with said management.

1.2 Detailed Study

This approach builds on the initial study by performing a more in-depth study of the current system. Another function of the detailed study is identifying potential users of the system. These users will exist on three different levels:

- Senior managers with profit responsibilities;

- The middle managers of the departments affected;
- The end-users, namely the users who will be working directly on the system;

After having identified these users, analysts will conduct interviews with them in order to ascertain their interests and expectations or requirements pertaining to the system to be developed. Data flow diagrams (DFDs) that extend beyond the boundaries of the current system will be constructed in order to identify the interfaces between various systems.

At the end of the detailed study the following should have been completed:

- a definition of the user community that will be using the new system(s);
- a logical model of the current system;
- a statement containing the increase in avoidable cost/revenue/improved server etc;
- An account of statutory/competitive pressures.

1.3 Defining and designing alternative solutions.

During this phase the current system is examined, existing problems are identified and alternative solutions are defined. Objectives relating to system functionality, which aids in management achieving organisational objectives, are defined. These objectives should be specific and measureable and are subsequently used to design a logical dataflow diagram of the new or desired system. Afterwards the methodology enters a design phase in which analysts as well as the designers cooperate in designing alternative implementation designs. These designs vary in addressed objectives. Alternatives should cover the three following categories:

1. Low budget, quick implementation;
2. Mid-budget, medium-term implementation period;
3. Higher-budget, long-term implementation.

Each alternative should contain a rough estimation of benefits, costs involved, hardware-, software requirements, as well as timescales for each implementation.

Lastly a report should be constructed which aids the relevant decision-makers in choosing the most applicable solution. This phase's report should contain:

- A DFD of the current system;
- A list of limitations of the current system, including benefits and cost estimates;
- A logical DFD of the system to be developed.

1.4 Physical design.

During this phase the following parallel activities will take place:

1. All the detail of the DFD must be produced.
2. The database or physical files will be designed.
3. The data stores have to be rationalized.
4. A modular hierarchy of the functions from the DFD should be designed.

Development Approach

STRADIS uses functional decomposition in its process modelling.

As stated by Gane and Sarson (1979). STRADIS is mainly concerned with the selection and organization of program modules and interfaces to help solve a predefined problem.

Process Model

STRADIS uses phases to aide in its development process. These phases focus on more on analysis than design; implementation is barely addressed in this systems development methodology. Furthermore a combination of techniques is used to illustrate and aid in each phase.

Tools and Techniques

As STRADIS' focus is on functional decomposition, the following techniques help in breaking complex systems into more manageable parts: Dataflow Diagrams (DFDs), Decision trees, and Decision tables.

2.3.2 Data-oriented systems development methodologies

Data-oriented systems development methodologies place a strong emphasis on developing accurate and complete data models of the system to be developed before proceeding with other aspects (Howard *et al.*, 1999).

Examples of blended systems development methodologies include:

- Structured System Analysis and Design Method (SSADM) – Eva (1994).
- Merise – Quang and Chartier-Kastler (1991).
- Information Engineering (IE) – Martin (1989).
- Welte ERP development – Welte (1999).

To better understand blended methodologies the following summary of IE can be reviewed:

Information Engineering (IE)

Development Method of IE

Information Engineering

2.1 Information strategy planning (ISP):

1.1 Current situation analysis: An overview of the business and its current position.

1.2 Executive requirements analysis: Managers can state their needs, objectives, and perceptions.

1.3 Architecture definition: Entails an overview of the information, an analysis of distribution, a definition of technical architecture, a definition of business systems architecture, and a definition of the information system organization.

1.4 Information strategy plan: Entails the establishment of business areas, the preparation of the information strategy plan itself, and the preparation of business evaluations.

2.2 Business area analysis:

The following tasks should be completed:

- Function and entity analysis. An analysis of entity types and their relationships, processes and dependencies, diagrams showing dependencies and entity models.
- Interaction analysis. Examines the interaction between the data and functions. Entity-relationship models are used to form an interaction matrix.
- Current system analysis. Models the existing system(s) in order to simplify the process of transitioning from one system to the next.
- Confirmation. Checks the results for accuracy, completeness, stability, etc.
- Planning for design. Defines which parts of the models are to be developed, evaluates the implementation process, and identifies reusable objects.

2.3 System planning and design

The following steps should be carried out during this phase:

- Preliminary data structure design. Attempt to convert the entity model to the structure of the chosen database system.
- System architecture design. Involves the mapping of processes to procedures in which interactions are highlighted by the use of data flow diagrams.
- Procedure design. Data navigation diagrams are established and action diagrams are designed.

- Confirmation. As in the business area analysis, the confirmation step checks for accuracy, stability and completeness.
- Planning for technical design. Defining implementation areas and preparing technical design plans.

2.4 Construction and cutover

Creation involves the following tasks:

- Systems generation. Constructing the database, files, generation models and the rest of the computing environment.
- System verification. Generate test data, performing system tests, and obtaining approval.

The tasks during the cutover are as follows:

- Preparation. Prepare the cutover schedule, hardware installations and training of users.
- Installation of new software.
- Final acceptance. Fully transferring to the new system and acceptance of the terms of agreements.
- Fan-out. Installing the system at all locations.
- System variant development. Identify and address construction where a location requires variance from the norm.

The following tasks ensure that the system is maintained and that changes in the business requirements are addressed:

- Evaluate system. Test performance and stability.
- Tune. Optimize system performance and stability.
- Maintenance. Correcting bugs and modifying the system as required.

Development Approach

IE was developed in the late 1970s, where Clive Finkelstein first used the term to describe a data modelling methodology (Martin, 1989), but James Martin went on to define IE as a generic class of methodologies, where the focus is strategic information systems and data modelling (Martin, 1991). Data modelling refers to the process of creating and analysing data models which are used to define data requirements by the business processes.

Process Model

Information Engineering employs the concept of parallel development. Parallel development refers to multiple processes being developed simultaneously.

Tools and Techniques

IE's main focus is data modelling, and presenting these diagrams to end-users or management is an effective way to communicate plans or requirements. As such the following techniques are used: Data-flow diagrams, Activity diagrams, Interaction diagrams, Entity Relationship Diagrams, Function/entity matrixes, bubble charts, user views, etc.

2.3.3 Object-oriented systems development methodologies

Object-oriented methodologies focus on development using objects; this includes their interaction with other objects, data and processes within a system (Avison & Fitzgerald, 2006:114). Object-orientation refers to the process of finding classes and objects, identifying structures and subjects, as well as defining attributes and services.

Examples of object-oriented methodologies include:

- Object-oriented analysis (OOA) – Coad and Yourdon (1991)
- Rational Unified Process (RUP) – Jacobsen *et al.* (1999)

To better explain object-oriented methodologies the following summary of RUP can be reviewed:

Rational Unified Process (RUP)

Development Method of RUP

Rational Unified Process

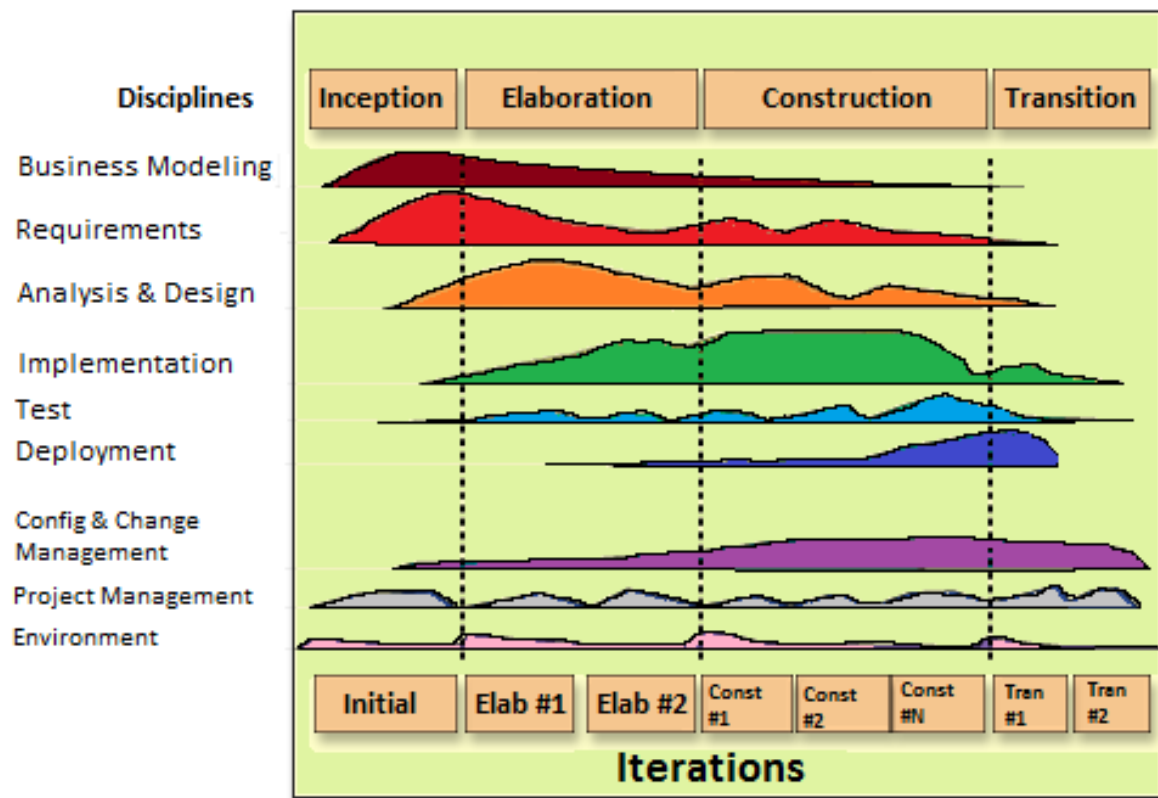


Figure 2.2 – Rational Unified Process – Process Model

- 3.1 The business modelling workflow - Establishes the context for the system being developed and the shape of the business in which the system will be implemented.
- 3.2 The requirements workflow - Establish with the project stakeholders what the system should do and why, define the project boundaries, and estimate the time-scales and costs involved.
- 3.3 The analysis and design workflow - Converts requirements from the requirements workflow into implementation specification.

- 3.4 The implementation workflow - Converts the designs into an implementation.
- 3.5 The test workflow - Tests and verifies the interaction of the components.
- 3.6 The deployment workflow - This workflow deploys the software to the users.
- 3.7 The configuration and change management workflow - Tracks and maintains the integrity of the project.
- 3.8 The project management workflow - Provides a framework for managing risks and software projects.
- 3.9 The environment workflow - Supports the project with relevant processes, tools, and methods in organization.

Development Approach

The Rational Unified Process was first called the Rational Object Process in 1997 and then renamed the Rational Unified Process in 1998 (Jacobsen *et al.*, 1999). Jacobsen went on to define RUP as “a full-fledged process able to support the entire software development life-cycle”. RUP utilizes the re-use of objects within an Object-Oriented environment.

Process Model

The Rational Unified Process uses iterative and incremental development, often referred to as spiral development. Within each iteration the development adds functionality and refines existing functions.

Tools and Techniques

The main techniques that RUP uses help in identifying objects and their re-usability. This includes: Business models, Object models, Unified Modelling Language, and Use cases.

2.3.4 Rapid systems development methodologies

“Rapid Systems Development” (RSD) is a structured approach with rigid limits on development time frames. The motto of RSD is faster, better, and cheaper” (Jain & Chandrasekaran, 2009:30). They continue to say that RSD combines rapidity and agility into the system development process.

Examples of Rapid development methodologies include:

- James Martin's RAD – Martin (1991)
- Dynamic Systems Development Method (DSDM) – Stapleton (2003)
- Extreme Programming (XP) – Beck and Andres(2005), Jeffries (2001), al-Tarawneh *et al.* (2011)
- Web Information Systems Development Methodology (WISDM) – Vidgen *et al.* (2002)

To better explain rapid development methodologies the following summary of XP can be reviewed:

Extreme Programming (XP)

Development Method of XP

Extreme Programming

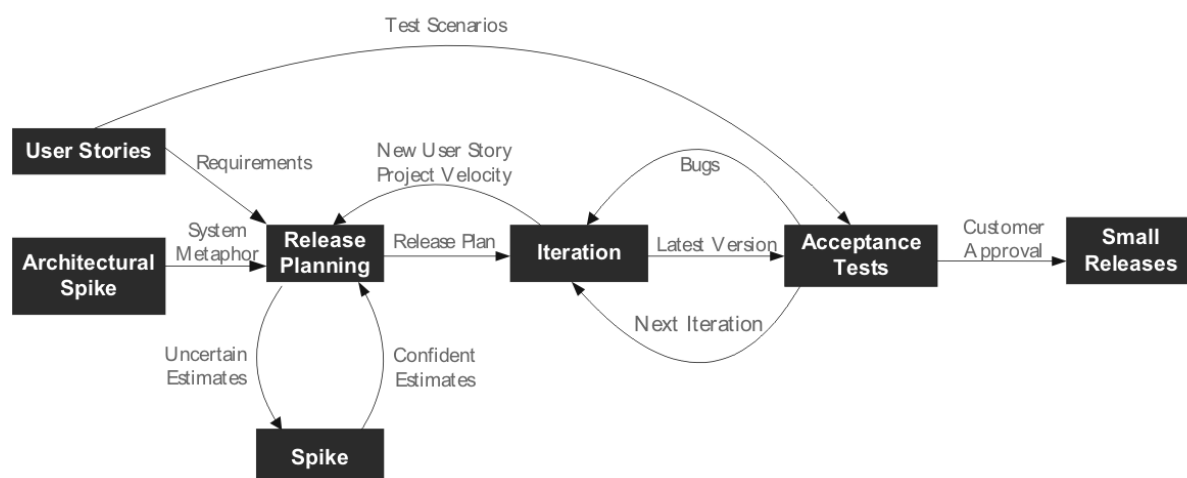


Figure 2.3 - XP Lifecycle (Cohn and Paul, 2001:1328)

XP uses phases to develop information systems, the following four stages can be identified Beck (2004):

- Planning - This phase relates to the scope of the project, the priority of each function, the members of the team, estimating financial costs, schedule release increments and an agreed quality level.
- Designing - Based on the principles of simplicity, courage and feedback, and enabling incremental change as described in the planning phase.
- Developing the code - This phase includes principles such as paired programming, testing using programmer and user data, retrieve rapid feedback, ensure the different tests work, and continuously integrate with already implemented code.
- Productionizing - Can be seen as part of the development phase, but during this phase the system as a whole is tested to ensure that it is ready for production.

Extreme programming encompasses the following twelve core principles:

- *Pair Programming*
- *Continuous Integration*
- *Collective Ownership*
- *Small Releases*
- *Testing*
- *Refactoring*
- *Metaphor*
- *Planning Game*
- *Coding Conventions*
- *Simple Design*
- *On-site Customer*
- *40 Hour Week*

Development Approach

XP focuses on the attempt to support quicker development of software. It consists of a series of principles rather than a step-by-step guide. Jeffries *et al.* (2001) define XP as an agile methodology with values of simplicity, communication, and feedback which serve as a software discipline.

Process Model

Extreme Programming uses iterative and incremental development; dividing development into phases while keeping documentation to a minimum.

Tools and Techniques

Various tools and techniques are used in XP - some of these include: Time boxing, Pareto principle, functional decomposition, MoSCoW rules, JAD work sessions, Prototyping (Architectural spikes), user stories, and paired programming.

2.3.5 People-oriented systems development methodologies

People-oriented methodologies focus on all the people who will be influenced by the project to be developed. This includes: Management, users, developers, stakeholders, etc.

Examples of People-oriented methodologies include:

- Effective technical and human implementation of computer-based systems (ETHICS) – Mumford (1995)
- KADS – Wielinga *et al.* (1993), Schreiber *et al.* (1993)
- CommonKADS – Schreiber *et al.* (1999), Tiwana (2000)

To better explain people-oriented methodologies the following summary of ETHICS can be provided:

Effective technical and human implementation of computer-based systems (ETHICS)

Development Method of ETHICS

Effective technical and human implementation of computer-based systems

Development steps as proposed by Mumford (1986) are as follows:

- 5.1 Why change?
- 5.2 System boundaries
- 5.3 Description of existing system
- 5.4, 5.5 and 5.6. Definition of key objectives and tasks
- 5.7 Diagnosis of efficiency needs
- 5.8 Diagnosis of job satisfaction needs
- 5.9 Future analysis
- 5.10 Specifying and weighting efficiency and job satisfaction needs and objectives
- 5.11 The organizational design of the new system
- 5.12 Technical options
- 5.13 The preparation of the detailed work design
- 5.14 Implementation
- 5.15 Evaluation

Development Approach

ETHICS embodies an ethical position (as the name implies). It serves as a participative approach to Information Systems Development (ISD) encompassing the need for a system to fit into social and organizational factors. ETHICS recognizes the interaction of technology and people and helps in producing systems which are both technically effective as well as socially acceptable which aids in raising job satisfaction (Mumford, 1995).

Process Model

ETHICS utilise steps which are completed in cycles.

Tools and Techniques

Tools and techniques are used to focus on determining social acceptance and integration, and as such some of the following techniques are used: Joint application development, Actor Analysis, Responsibility Matrices, etc.

2.3.6 Organizational-oriented systems development methodologies

Examples of Organizational-oriented methodologies include:

- Soft Systems Methodology (SSM) – Checkland (1991)
- Information systems work and analysis of change (ISAC) – Lunderberg *et al.* (1982)
- Process Innovation (PI) - Davenport (1993)
- Projects in controlled environments (PRINCE) – Bentley (2009)
- Renaissance – Warren (1999)

To better explain Organizational-oriented methodologies the following summary of SSM can be examined:

Soft Systems Methodology (SSM)

Development Method of SSM

Soft Systems Methodology

SSM consists of seven main stages, these stages are as follows:

- 6.1 The problem situation: unstructured. Attempts to determine the problem situation by assessing as many as possible views from the people involved.
- 6.2 The problem situation: expressed. Attempt to express the problem situation in some formal way as determined in Stage 1.
- 6.3 Root definitions of relevant systems. Explore possible relevant systems to determine which one is the most useful.
- 6.4 Building conceptual models. After the root definitions have been properly defined, and the problem owners as well as problem solvers are satisfied with these definitions, a conceptual model can be developed by using these root definitions.
- 6.5 Comparing conceptual models with reality. Using rich pictures alongside the conceptual models, these can be compared to the problem situation in order to lead to possible recommendations.
- 6.6 Assessing feasible and desirable changes. Review the proposed changes from Stage 5. Attempt to draw up proposals for the changes based on both desirability and feasibility.
- 6.7 Action to improve the problem situation. Recommend actions which will help alleviate the problem situation.

Development Approach

The Soft Systems Methodology's approach aims to develop a system for the organization as a whole rather than its development being an isolated function Checkland (1981).

Process Model

The process model used in the Soft Systems Methodology consists of phases which are iteratively developed.

Tools and Techniques

This implies the tools and techniques employed by SSM focus on assessing the system and organization as a whole; and it includes: Rich pictures, root definitions, and conceptual models.

2.4 Systems development methodology comparisons

Various frameworks exist which can be used to compare systems development methodologies. One such example is the framework discussed in Chapter 2.3. The framework by Huisman and Iivari (2006:32) identifies four core aspects by means of which a systems development methodology can be uniquely identified. This includes a systems development method; philosophical approach; process model; and techniques.

Another example of such a framework is the framework used by Fitzgerald and Avizienis (2006:598). The latter's framework is as follows:

- 1) Philosophy. The principle, or set of principles, which define the systems development methodology.
 - a) Paradigm. The manner in which a problem is approached.
 - b) Objectives. The main goals which the systems development methodology aims to achieve.
 - c) Domain. The situations in which the systems development methodology can normally be applied.
 - d) Target. The environment, types of problems, or types of organisation to which the systems development methodology is applicable.
- 2) Model. An abstract of the systems development methodology's view of the world, normally factors which are important to the information system or the organisation.
- 3) Techniques and tools. Tools which the systems development methodology applies to aid in developing the information system.
- 4) Scope. Indicates which stages of the systems development life cycle a systems development methodology covers.

- 5) Outputs. Deliverables produced at each stage and the nature of the final deliverable in particular.
- 6) Practice. The systems development methodology background, user base, participants, and the required skill levels.
- 7) Product. The end-product delivered to the clients. Can include documentation, software, training sessions, telephone help-line etc.

In this research the framework identified by Huisman and livari (2006:32) is used to compare a basic set of methodologies. The said framework provides an easy and effective means to differentiate between different categories of systems development methodologies whilst also providing a clear summary of each systems development methodology. These summaries can then be used to compare the applicable systems development methodologies in order to review the advantages and disadvantages which each offer as well as the appropriate situation wherein each can/should be implemented.

2.4.1 Methodology comparisons

To understand what these methodologies share, we look at the similarities between them and how they influence project development. A summary in tabular form can be viewed in Table 2.2.

Table 2.2 – Methodology comparison

	Process-oriented Methodologies	Blended Methodologies	Object-oriented Methodologies	Rapid development Methodologies	People-oriented Methodologies	Organizational- oriented Methodologies
Methodology	STRADIS	IE	RUP	XP	ETHICS	SSM
Systems Development Method	Structured analysis, design and implementation of information systems	Information Engineering	Rational Unified Process	Extreme Programming	Effective technical and human implementation of computer-based systems	Soft Systems Methodology
Philosophical Approach	-Functional decomposition -Process Modelling	-Strategic Information Systems -Data Modelling	-Object-orientation	-Rapid Application Development -Agile systems approach -Object orientation	-People-oriented -Social-technical	-System theory
Process Model	-Phases (mainly focuses on analysis, less on development, and almost no focus on implementation)	-Parallel development	-Iterative and incremental -Spiral development	-Iterative development -Incremental development -Phases	-Steps carried out in cycles	-Iterative development in phases

Tools & Techniques	-Dataflow diagrams -Data structure diagrams -Decision trees -Decision tables -Structured English -Normalisation	-Entity modelling -Normalisation -Entity life cycle -Decision trees -Decision tables -Structured English -Action diagrams -Critical success factors	-Unified modelling language -Class diagrams -Use case diagrams -Interaction diagrams -Sequence diagrams -State chart diagrams -Activity diagrams -Pert diagrams -Gantt diagrams	-Functional decomposition -MoSCoW-rules -User stories -Pareto principle -JAD-work sessions -Paired programming -Prototyping	-Actor analysis -Joint application development	-Rich pictures -Root definitions -Conceptual models
-------------------------------	--	--	---	---	---	---

2.4.1.1 Similarities

- Both STRADIS and IE use decision tables, decision trees and structured English to develop information systems. These tools aid in planning the project, deciding between various alternatives and structuring the development in a logical and rational order.
- All of these methodologies use developmental steps/stages/phases to develop information systems. In IE and XP different phases can be developed during the same time. In STRADIS and ETHICS the steps have a certain order in which they should take place; these steps cannot be worked on during the same period or executed in any other sequence. With RUP and SSM the development phases are iterative, which means that they can be repeated until the desired results are achieved. At the end of an iteration the system's features should have increased or they should have been further refined.
- All these methodologies focus on the end user. This focus helps in developing a system that fulfils the end user's needs and ensures that the system will actually be used to address certain requirements.

The differences between these previously mentioned methodologies can now be discussed. These differences can provide some insight into why a certain methodology will be better suited to the development of a certain information system.

2.4.1.2 Differences

- All these methodologies are based on different philosophical approaches. These philosophical approaches determine what the methodology attempts, achieves or focuses on. For example, STRADIS focuses on functional decomposition, whereas SSM focuses on the system as a whole, "One tenet of systems thinking is that the whole is greater than the sum of its parts" as stated by Fitzgerald and Avison (2006:507).
- Methodologies like ETHICS and RUP differ in the way that they deal with the execution of their development steps. In ETHICS and STRADIS the development steps have a certain order in which it should be executed; these steps are generally only executed once, differing from methodologies such as SSM and

RUP where the development steps are iterative and incremental. Some development steps can be repeated a number of times.

- Methodologies such as IE and STRADIS differ in the way that they approach their development. IE focuses on the development of data and information, whereas STRADIS focuses on the processes.

2.4.2 Methodology comparisons conclusion

It is clear that methodologies differ from each other. Process Maturity Models require that organizations use SDMs during their systems development process. The question is then – which SDMs are better suited to meet this requirement as well as help the organization in developing a better system in a shorter time? This research will attempt to clarify whether the use of SDMs as certification requirement effectively helps organizations develop their information systems.

2.5 Introduction to PMMs

To review: the goal of this study is to determine the relationship (if any) between the use and effectiveness of Systems Development Methodologies and Process Maturity Models such as the Capability Maturity Model Integration (CMMI) and ISO 90003. In order to form a comprehensible understanding of these key concepts the following definitions concerning these processes and concepts are given:

Demir and Kocabaş (2010): “A maturity model provides a systematic framework for carrying out benchmarking and performance improvement.”

Prikladnicki and Audy (2010:780) define Process Models as a set of practices or a set of stages (or steps) which were successfully followed by project teams, individuals or organizations. These practices were documented as successful practices capable of being adopted by other peers.

Furthermore Chrissis *et al.* define Capability as the range of expected results which can be achieved by following a process, or the predictability of the process and its outcomes. The authors continue by defining Maturity as the growth in the process capability. It can be viewed as a well-defined evolutionary path which can be followed to achieve a mature process; each

maturity level provides a layer in the foundation for continuous process improvement and each level obtained serves as a layer in the foundation for continuous process improvement. For each level of a maturity framework achieved, it results in an increase in the process capability.

In the current working environment great pressure is placed on organizations to be certified in some kind of Process Maturity Model (Fitzgerald, 1998:317). This ensures that information systems to be developed will conform to a set of quality standards as well as include a set of development standards and methods. Maturity models like ISO 9000-3 and CMMI require some kind of standardized development process which in most cases forces organizations to adapt or implement one (or more) of the many Systems Development Methodologies available today. This in turn causes organizations to be forced into the usage of development tools or processes of which they might have little to no knowledge of and the implementation thereof consequently leads to failed projects or inadequate systems.

Some of the possible Process Maturity Models include:

- Project Management Process Maturity (PM)² – Kwak & Ibbs (2002)
- MIS-PyME - Díaz-Ley *et al.* (2010)
- Goal Question Metric – Basili *et al.* (1994)
- Goal-Driven Software Measurement – Basili *et al.* (2009)
- GQ(I)M – Goethert (2003)
- CMMI - Carnegie Mellon University (2011)
- ISO 9000 (ISO/IEC 90003:2004) – International Organization for Standardization (2010)
- TickIT – Bamford & Deibler (2003)

In this chapter we are specifically focusing on ISO 90003:2004, which is the “latest” version of the ISO 9000-3 standard. Whenever the term ISO 9000-3 is used in this dissertation it refers to the ISO 90003:2004 version, its certification requirements (ISO 9001:2008), and CMMI. In order to obtain an understanding of what each model entails in regards to requirements, as well as the techniques used to ensure project success, each model will be discussed in terms of its history, usage, and critique against its use.

2.6 ISO 9000

In this section ISO 9000 is discussed. An overview of where, why, and how it originated is provided; furthermore ISO 9000's certification requirements (ISO 9001:2008) as well as observed advantages and disadvantages are reviewed.

2.6.1 The origin and history of ISO 9000

ISO 9000 is a set of quality standards used mainly in Europe to ensure the development of quality systems. The ISO 9000 family of certification standards is used to address "quality management", whereas the ISO 9000-3 certification was specifically designed for usage in the developing computer systems. ISO 9001:2008 is the latest standard that "provides a set of standardised requirements for a quality management system, regardless of what the user organization does, its size, or whether it is in the private or public sector" (Landry, 2011). The ISO 9001:2008 is the only standard in the ISO 9000 family against which organizations can be certified (International Organization for Standardization, 2011).

The ISO 9000 family of standards is mainly used in European countries and Japan, while CMMI is mainly used in the United States of America (Lee *et al*, 2009).

In short, ISO 9000 examines how the company involved solves the following issues:

- The quality requirements set by the customer,
- Applicable regulatory requirements,
- Enhanced customer satisfaction, and
- Achievement of continual improvement of its performance in the pursuit of these objectives.

Numerous information technology projects are being developed and some of these projects are not considered to be of adequate quality if measured against a set of standardized principles, such as staying within the project budget, schedule, and fulfilling user requirements. In spite of these factors, many organizations still consider themselves to be unique; this is where ISO 9001:2008 lays down a set of requirements which a quality system should meet. ISO 9001:2008 does not, however, dictate how these requirements should be met in any

particular organization. This leaves organizations with a greater scope and flexibility for implementations in different business cultures and business sectors.

According to the International Organization for Standardization (ISO 9000 Essentials, 2011:1) the following can be done to help ensure that projects will most likely be of a high quality:

1. **The standard requires the organization itself to audit** its ISO 9001:2008-based quality system to verify that it is managing its processes effectively - or to make certain that it is fully in control of its activities.
2. In addition, **the organization may invite its clients to audit** the quality system so that they can be confident that the organization is capable of delivering products or services that will meet their requirements.
3. Lastly, the organization may engage the services of an **independent quality system certification body** to obtain an ISO 9001:2008 certificate of conformity. This last option has proved extremely popular in the marketplace because of the perceived credibility of an independent assessment.

"The organization may thus avoid multiple audits by its clients, or reduce the frequency or duration of client audits. The certificate can also serve as a business reference between the organization and potential clients, especially when supplier and client are new to each other, or far removed geographically, as in an export context" (ISO 9000 Essentials).

2.6.2 ISO 9000 Certification requirements

The ISO 90003:2004 Software Standardization guidelines were summarised by the Praxiom Research Group Limited (ISO IEC 90003:2004 Software Standard) from technical jargon used in the original ISO specifications to a more easily understandable English version; their version is as follows:

2.6.2.1 Systematic Requirements and Guidelines

- 1.1 Establish a quality management system for software products. This involves developing a quality system for the software products and the related services. The processes within the quality system should also be described. Lastly the quality management system should be implemented, reviewed and improved if possible.

- 1.2 Document your software-oriented quality system. Develop the documents which describe the quality system; create a system manual; control these quality management documents; and maintain the quality management system records.

2.6.2.2 Management Requirements and Guidelines

- 2.1 Support quality. Emphasise the importance of quality; develop a system to manage Quality assurance; implement the quality assurance system; and improve the quality assurance process.
- 2.2 Focus on your customers. Accurately identify customer requirements; meet these requirements; and enhance customer satisfaction.
- 2.3 Establish a quality policy. Define a quality policy and manage this policy.
- 2.4 Perform quality planning. Establish quality objectives and plan the quality management system.
- 2.5 Control your quality system. Define responsibilities and authorities; appoint an appropriate management representative. Lastly ensure that an internal communications process is established and implemented.
- 2.6 Perform management reviews. Review the quality management system; examine input obtained from management reviews and generate management review outputs.

2.6.2.3 Resource Requirements and Guidelines

- 3.1 Provide quality resources. Identify resources required by the quality management system; provide these resources in order to support the quality management system.
- 3.2 Provide quality personnel. Ensure that personnel have the required knowledge, abilities, training, etc. Define levels of competence and ensure that employees receive the required training and awareness programmes.
- 3.3 Provide quality infrastructure. Identify the infrastructure required to develop the software; identify tools used in software management; provide the infrastructure identified along with the tools required to manage the software development.

Maintain the above-mentioned infrastructure as well as the tools required in the software development management.

- 3.4 Provide a quality environment. Identify; implement; and manage the required working environment.

2.6.2.4 Realization Requirements and Guidelines

- 4.1 Control software products and realization planning. Identify the product realisation processes; this includes quality objectives, requirements, etc., afterwards develop the said realisation process. Life-cycle models can be used to plan the required work; afterwards software quality planning should be carried out.
- 4.2 Control customer processes. Identify the customer's software product requirements; identify other requirements related to the customer; review these requirements and identify potential organisational software product concerns. Evaluate the risks identified by the said requirements; appoint a customer representative; communicate with the customers; communicate relevant information to the customers during development; and lastly communicate with customers during the implementation and maintenance.
- 4.3 Control software design and development. Plan the software design and development; plan review, verification, and validation activities. Define inputs which will be required during the design and development process; generate software and development outputs; carry out reviews of the said design and development outputs, afterwards verify that the design and development activities were carried out correctly. Validate these results and carry out software testing. Manage and design any changes required to the system.
- 4.4 Control your purchasing function. Ensure that acquired products meet the requirements; control the software purchasing process along with purchased parts and components. All of these purchases should be documented and verified by means of acceptance testing etc.

- 4.5 Manage production and service provision. Control the software build, release, replication, etc.; validate product as well as service provision; identify and track the products. Property supplied by the customers should be protected and the organisation's own property should be preserved.
- 4.6 Control monitoring devices. Identify needs in regards to monitoring and measurement devices; select the applicable devices; calibrate the said devices; protect these devices and validate them before implementation.

2.6.2.5 Remedial Requirements and Guidelines

- 5.1 Carry out remedial processes. Plan how to monitor, measure, and analyse processes in order to demonstrate conformity, maintain quality and continually improve the quality of the management system. Implement the said remedial processes.
- 5.2 Monitor and measure quality. Customer satisfaction should be monitored; regular internal audits should be performed; the applied quality processes should be monitored and measured as well as product characteristics in order to effectively measure and improve on development quality.
- 5.3 Control your non-conforming software products. Establish a procedure which will be used for non-conforming software products. These non-conforming products should be identified and corrected. Afterwards these products should be verified to confirm that they now conform to requirements.
- 5.4 Analyse quality information. Define the needs of the organisation to ensure quality information. Collect and measure the suitability as well as the effectiveness of the quality system in order to provide quality management information.
- 5.5 Take required remedial actions. The quality management system should be maintained. Non-conformities should be corrected whilst other potential non-conformities should be identified and avoided.

2.6.3 Advantages of being certified in ISO 9000

As an organization evolves it is important to keep clients satisfied in order to ensure the support of current clients as well as to gain potential clients. In order to keep clients satisfied, the organization should strive to meet the requirements of their current and future clients. ISO 9001:2008 provides a standard for taking a systematic approach to managing the organization's processes in order to consistently produce products which fulfils the client's requirements. These standards have been proven to be effective in most of its applications.

According to Craig (1994:38) the benefits of implementing the ISO 9000 series of standards fall into two categories. They are the following:

- Tangible benefits:
 - Improved product quality, with less resources being spent on reworks for nonconforming products.
 - Improved delivery performance.
 - Reduced cost.
- Intangible benefits:
 - An increased organizational commitment to quality.
 - Registration for ISO 9000 involves every employee. Because of this, a successful registration creates a strong sense of unity between employees.
 - Customer focus shifts from auditing the organizations system to establishing and maintaining strong customer-supplier relationships. This gives an organization a distinct competitive advantage.
 - Customers and employees gain confidence in quality management systems.
 - Organized record-keeping and standardized procedures create a better environment for process analysis, and provide a basis for further improvements.

Being certified in ISO 9000 also ensures that, at the very least, an organisation's maintenance processes are properly defined (Taylor & Wood-Harper, 1996) and that it can assist in producing high quality software, reducing development time and cost, and increasing productivity (Butler, 1995; Pitterman, 2000; Yamamura, 1999; Niazi *et al.*, 2005).

2.6.4 Criticisms against the implementation of ISO 9000

As with various other SPI techniques, criticism exists with regard to the ISO 9000 standards as well. Limited or incomplete implementation of the said SPI technique is often the main culprit when unwanted results are obtained. Seeing as ISO 9000 is another form of SPI, it shares similar criticisms with CMMI (discussed in 2.7.4).

To summarise, the main criticisms against ISO 9000 is as follow (Goldenson & Herbsleb, 1995; Krasner & Ziehe, 1995; Loken & Skramstad, 1995; Stelzer *et al.*, 1996; Stelzer *et al.*, 1997; Stevenson & Barnes, 2002; Coleman & O'Connor, 2008):

- Too costly. The process of implementing certification requirements as well as the certification in itself is believed to be too expensive.
- Experts also believe ISO 9000 certification is a pursuit of quality certification rather than a pursuit of quality.
- Brown (1994) specifically speculates that the only organisations guaranteed to profit from ISO 9000 implementations are the ones qualified to do the audits and issue the certificates.
- Requires a lot of paperwork and also adds so many extra structures, some believe it interferes with newer and better ways of operating
- Individual countries' standardisation organisations are left to regulate and implement standards. No concrete guidelines exist and the amount of work required for certification can vary according to the registrar's country's regulations.
- The ISO 9000 standard in itself is not industry-specific. Some critics claim that this results in failure to address specific issues.

2.7 Capability Maturity Model Integration (CMMI)

Capability Maturity Model Integration provides a set of guidelines to help organizations improve their project development performance. CMMI can be used to guide process improvement across a project, a division, or an entire organization. The model aids in integrating rationality separate or organizational functions, "set improvement goals and priorities, provide guidance for quality processes, and provide a point of reference for

appraising current processes" (CMMI - Software Engineering Institute, 2011; al-Tarawneh *et al.* 2011).

2.7.1 The origins and historical use of CMMI

CMMI is spreading worldwide and being adopted in continents such as North America, Europe, Asia, Australia, South America, and Africa. This has resulted in the SEI's commitment to CMMI.

CMMI can be used in three different areas of interest (CMMI - Software Engineering Institute, Carnegie Mellon):

- Product and service development (The CMMI for Development model or CMMI-DEV)
- Service establishment, management, and delivery (The CMMI for Services model or CMMI-SVC)
- Product and service acquisition (The CMMI for Acquisition model or CMMI-ACQ)

In the Version 1.2 Overview of CMMI, the guidelines introduce the CMMI concept that can help organizations make decisions regarding its process improvement plans. These guidelines can also be used to inform other employees about CMMI.

Different CMMI models exist, and these models are collections of best practices that users can compare to their organization's own best practices to help guide their improvement. An appraisal refers to a formal comparison of a CMMI model to an organization's processes. The Standard CMMI Appraisal Method for Process Improvement (SCAMPI) uses the best ideas from several process improvement appraisal methods.

CMMI, originally referred to as the CMMI project, was developed at the Carnegie Mellon University by American government officials and a range of information technology experts working in cooperation with the Software Engineering Institute (SEI). The office of the US Secretary of Defence (OSD) and the American National Defence Industrial Association were the main sponsors of the project.

CMMI is the successor of the Capability Maturity Model (CMM) or software CMM. The CMM was developed between 1987 and 1997. In 2002, CMMI Version 1.1 was released followed by Version 1.2 in August 2006.

2.7.2 Understanding CMMI levels an overview

"Levels are used in CMMI to describe an evolutionary path recommended for an organization that wants to improve the processes it uses to develop and maintain its products and services" (CMMI® for Development, Version 1.2 p. 29).

CMMI supports two improved paths: Continuous and Staged. The Continuous path enables organizations to incrementally improve processes focussing on individual process areas identified by the organization as important for its immediate business objectives. The other path (Staged) helps organizations to improve processes significantly by incrementally addressing successive sets of process areas.

The five maturity levels include (Whitten *et al.*, 2001:77):

- Level 1- Initial: This level is sometimes called anarchy, chaos or ad hoc. The system development process follows no prescribed path. Each person in the development team uses his/her own tools and methods. This development process is sporadic, unpredictable and not repeatable. The success of the project is dependent on the skill and the experience of the development team. Documentation is not consistent from one phase of the project to the next, creating problems for the people using and maintaining the system through its lifetime. This level of maturity is plagued with cost and time overruns.
- Level 2 – Managed: In this level of maturity there are some project management processes and systems in place to track cost, scope, schedules and functionality. The focus of this level is placed on project management and not systems development. Success or failure is still dependent on the skill and experience of the development team; however, the development team makes an effort to repeat their previous successes. Cost and time overruns are still commonplace.
- Level 3 – Defined: A standard system development method/methodology has been purchased or developed and its use has been integrated throughout the entire information system development department. All projects use an adapted version of this methodology to develop and maintain information systems and software. With the use of standardized processes, project results are consistent and of higher quality. Improvements in documentation make maintenance and

use of these systems more effective. The processes are stable, predictable and repeatable.

- Level 4 - Quantitatively Managed: Measurable goals for quality and productivity are established. Detailed measures of the system development methodology and product quality are routinely collected and analysed. This makes management more proactive than reactive to problems in systems development projects. Thus, when unexpected problems occur, management can adjust the development process based on predictable and measurable impacts.
- Level 5 Optimizing: The systems development process is continually monitored and improved based on measures and data analysis gathered from the managed level as well as project management feedback. This can involve the changing of technologies, tools and methods used in the development process. Lessons learned from previous projects are shared and used in the organization with the goal of eliminating inefficiencies in the system development process without compromising the quality of the product. In summary, the organization is doing continuous system development improvements.

2.7.2.1 CMMI model framework

Depending on the CMMI constellation (acquisition, services, and development) used, the process areas it contains will vary. The areas that will be covered by the organization's processes are known as the key process areas. These process areas are summarized in Table 2.3.

Table 2.3 - CMMI for Development, Version 1.3(Carnegie Mellon, 2008) key process areas

Abbreviation	Name	Area	Maturity Level
1.CM	Configuration Management	Support	2
2.MA	Measurement Analysis	Support	2
3.PMC	Project Monitoring and Control	Project Management	2
4.PP	Project Planning	Project Management	2
5.PPQA	Process and Product Quality Assurance	Support	2
6.REQM	Requirements Management	Project Management	2
7.SAM	Supplier Agreement Management	Project Management	2
8.DAR	Decision Analysis and Resolution	Support	3
9.IPM	Integrated Project Management	Project Management	3
10.OPD	Organizational Process Definition	Process Management	3
11.OPF	Organizational Process Focus	Process Management	3
12.OT	Organizational Training	Process Management	3
13.PI	Product Integration	Engineering	3
14.RD	Requirements Development	Engineering	3
15.RSKM	Risk Management	Project Management	3
16.TS	Technical Solution	Engineering	3
17.VAL	Validation	Engineering	3
18.VER	Verification	Engineering	3
19.OPP	Organizational Process Improvement	Process Management	4
20.QPM	Quantitative Project Management	Project Management	4
21.CAR	Casual Analysis and Resolution	Support	5
22.OPM	Organizational Performance Management	Process Management	5

Maturity Level 2 – Managed

CM - Configuration Management - The purpose of Configuration Management (CM) is to establish and maintain the integrity of produced products using configuration identification, configuration control, configuration status accounting and configuration audits.

MA - Measurement and Analysis - The purpose of Measurement and Analysis (MA) is to develop and sustain a measurement capability used to support management information needs.

PMC - Project Monitoring and Control - The purpose of Project Monitoring and Control (PMC) is to provide an understanding of the project's progress to ensure appropriate corrective actions can be taken when the project's performance deviates significantly from the plan.

PP - Project Planning - The purpose of Project Planning (PP) is to establish and maintain plans that define project activities.

PPQA - Process and Product Quality Assurance - The purpose of Process and Product Quality Assurance (PPQA) is to provide staff and management with objective insight into processes and associated work products.

REQM - Requirements Management - The purpose of Requirements Management (REQM) is to manage the requirements of the project's products and product components and to ensure alignment between those requirements and the project's plans and produced products.

SAM - Supplier Agreement Management - The purpose of Supplier Agreement Management (SAM) is to manage the acquisition of products and services from suppliers.

Maturity Level 3 – Defined

DAR - Decision Analysis and Resolution - The purpose of Decision Analysis and Resolution (DAR) is to analyse possible decisions using a formal evaluation process that assesses identified alternatives against established criteria.

IPM - Integrated Project Management - The purpose of Integrated Project Management (IPM) is to establish and manage the project and the involvement of relevant stakeholders according to an integrated and defined process that is tailored from the organization's set of standard processes.

OPD - Organizational Process Definition - The purpose of Organizational Process Definition (OPD) is to establish and maintain a usable set of organizational process assets, work environment standards, and rules and guidelines for teams.

OPF - Organizational Process Focus - The purpose of Organizational Process Focus (OPF) is to plan, implement, and deploy organizational process improvements based on a thorough understanding of current strengths and weaknesses of the organization's processes and process assets.

OT - Organizational Training - The purpose of Organizational Training (OT) is to develop skills and knowledge of people so that they can perform their responsibilities effectively and efficiently.

PI - Product Integration - The purpose of Product Integration (PI) is to assemble the product from its components, ensure that the product, as integrated, behaves properly (namely, possesses the required functionality and quality attributes), and deliver the product.

RD - Requirements Development - The purpose of requirements Development (RD) is to elicit, analyse, and establish customer, product, and product component requirements.

RSKM - Risk Management - The purpose of Risk Management (RSKM) is to identify potential problems before they occur so that risk mitigation activities can be planned and implemented as needed throughout the lifecycle of the product or project to alleviate adverse impacts on achieving objectives.

TS - Technical Solution - The purpose of Technical Solution (TS) is to select, design, develop, and implement solutions to meet the necessary requirements. Solutions, designs, and implementations encompass products, product components, and product-related lifecycle processes either singly or in combination as appropriate.

VAL – Validation - The purpose of Validation (VAL) is to demonstrate that a product or product component fulfils its intended use when placed in its intended environment.

VER – Verification - The purpose of Verification (VER) is to ensure that selected work products meet their specified requirements.

Maturity Level 4 - Quantitatively Managed

OPP - Organizational Process Performance - The purpose of Organizational Process Performance (OPP) is to establish and maintain a quantitative understanding of the

performance of selected processes in the organization's set of standard processes in support of achieving quality and process performance objectives, and to provide process performance data, baselines, and models to quantitatively manage the organization's projects.

QPM - Quantitative Project Management - The purpose of Quantitative Project Management (QPM) is to quantitatively manage the project so that it achieves its established quality and process performance objectives.

Maturity Level 5 – Optimizing

CAR - Causal Analysis and Resolution – The purpose of Causal Analysis and Resolution (CAR) is to identify causes of selected outcomes and take action to improve process performance.

OPM - Organizational Performance Management - The purpose of Organizational Performance Management (OPM) is to proactively manage the organization's performance to meet its business objectives.

Only some processes are present in all CMMI constellations, these include:

Abbreviation 	Name 	Area 	Maturity Level 
CM	Configuration Management	Support	2
MA	Measurement and Analysis	Support	2
PMC	Project Monitoring and Control	Project Management	2
PP	Project Planning	Project Management	2
PPQA	Process and Product Quality Assurance	Support	2
REQM	Requirements Management	Project Management	2
DAR	Decision Analysis and Resolution	Support	3
IPM	Integrated Project Management	Project Management	3
OPD	Organizational Process Definition	Process Management	3
OPF	Organizational Process Focus	Process Management	3
OT	Organizational Training	Process Management	3
RSKM	Risk Management	Project Management	3
OPP	Organizational Process Performance	Process Management	4
QPM	Quantitative Project Management	Project Management	4
CAR	Causal Analysis and Resolution	Support	5
OPM	Organizational Performance Management	Process Management	5

Fig 2.4 - CMMI Core Process Areas (Carnegie Mellon, 2009)

2.7.2.2 CMMI Appraisals

An organization cannot be certified in CMMI. Instead, an organization is appraised. The organization can receive a maturity level rating (between 1 and 5) or a capability level achievement profile depending on the type of appraisal.

Organizations can measure their progress in this maturity model by conducting an appraisal. They can normally conduct an appraisal to fulfil one of the following reasons:

1. To fulfil contractual requirements of clients.
2. To help identify other process areas which should receive more effort and to measure how well the organization's processes compare to CMMI.
3. To be able to take market advantage and advertise or inform external clients of how well the organization's processes compare to the CMMI best practices, which in turn results in more successful projects.

Three classes for CMMI appraisals exist, viz. A, B, and C. These classes are used to focus on identifying improvement opportunities and comparing the organization's processes to that of the CMMI's best practices. A class A appraisal is a more formal appraisal and is the only class which can be used to deliver a level rating result. The results of an appraisal may be published (as long as the organization approves the action) on the CMMI web site of the SEI: Published SCAMPI Appraisal Results. SCAMPI also supports SPICE, which is the conduct of the ISO/IEC 15504 standards.

2.7.3 Advantages of being certified in CMMI

Implementing CMMI offers various opportunities for improving software development processes. In this section we examine some of these benefits.

Some of the benefits obtained from using CMMI include (Mesquida *et al.*, 2012; Sulayman *et al.*, 2012; Wendler, 2012; Ashrafi, 2003; Li Eldon *et al.*, 2002):

- The activities of the organization are explicitly linked to the organization's business objectives.

- Customer's expectations of the product or service are met by increasing your visibility into the organization's activities.
- The organization gets the opportunity to learn from new areas of best practise (for example measurement, risk, etc.)
- Consistent and repeatable processes can be followed and maintained.
- Technical skills of staff members are improved the use of technology is maximised.
- Organisations can develop, maintain and deliver higher quality products.
- Focus on quality improvement as well as time and cost reduction.
- Higher quality products lead to higher customer satisfaction.
- Increased organisational flexibility

In a study by Staples and Niazi (2005), the authors investigated the motivations for adopting CMM-based software process improvement. This can also be interpreted as the perceived benefits of CMMI or any other SPI model. Their results indicated that 80% of organisations participating in their study found at least one of the following reasons to motivate CMM-based SPI:

- Software quality (66% of respondents indicated this as a reason)
- Development time (55% of respondents indicated this as a reason)
- Development cost (55% of respondents indicated this as a reason)
- Productivity (41% of respondents indicated this as a reason)
- Process visibility (30% of respondents indicated this as a reason)
- Customer demands (21% of respondents indicated this as a reason)
- Market advantage (20% of respondents indicated this as a reason)
- Software Process Improvement (18% of respondents indicated this as a reason)

2.7.4 Criticisms against the implementation of CMMI

As with most techniques employed to improve development processes or success, various criticisms exist against CMMI implementation. Mostly this critique exists due to incorrect application or incomplete implementations. In this section we examine some of the critique against the implementation CMMI.

In a study by Niazi and Ali Babar (2009), the authors state that one of the biggest factors militating against the implementation of CMMI is the sheer amount of time and resources

invested in its implementation and that even after this investment there is no guarantee that it will live up to the its expectations. They authors continue by stating that the implementation process increases even further in difficulty for smaller organisations, thus limiting the benefits even further. In another article Niazi and Staples (2007) express their concerns in regards to the implementation of CMMI in certain organisations, especially Small and Medium Enterprises (SMEs).

In a very detailed study by Staples *et al.* (2007) the authors focused explicitly on the reasons why organisations do not implement CMMI, and they state that prior to their study no research had been done on that specific subject. To summarize, the top reasons they found why organisations choose not to adopt/implement CMMI certification were:

- The organisations felt they were too small to fully commit to the implementation of CMMI;
- The organisations did not have sufficient time and resources available to implement CMMI;
- The organisations were already using some other form of software process improvement techniques.

Critics have also argued that little or no evidence exists to prove the value of process improvement techniques such as CMMI, while others have expressed their concerns about the validity of the results (Bollinger & McCowan, 1991; Fayad & Laitinen, 1997; Gray & Smith, 1998; Jungh, 2005).

2.8 CMMI and Systems Development Methodologies integration

Little to no research has been done on the particular subject of SDM integration within Process Maturity Models and how effective their use is. A study by Niazi *et al.* (2005) on the subject of a Process Maturity Model which implements Software Process Improvement (SPI) has shown that practitioners are more concerned about awareness activities and how to implement these SPI activities than the details of the physical activities themselves. This indicates that they do indeed see the process of improving on development activities as a long-term benefit. Sun and Liu (2010) also identify that a methodology is needed to “transform the CMMI activities into a set of actions which are detailed enough to be followed by software engineers”. In a study by al-Tarawneh *et al.* (2011) the authors identify the distinction between large and small software development firms and how they approach Process Maturity Models. The authors continue by identifying Extreme Programming (XP) as an effective systems development methodology to

implement in small organisations when attempting to conform to standards set out in the CMMI; this will then in turn lead to software development process improvement. In another study by Arias *et al.* (2012), the authors identify the best practices from the Rational Unified Process along with the PMI framework to guide project managers to focus on key factors which will raise changes of delivering a successful project while also stabilising their software development process.

From this it is clear that implementing a software development methodology aids in raising process maturity, which in turn raises the quality of the development process as well as the developed product. This forms the core of this research, as awareness of improving process maturity continues to rise. The continued process of improving on development processes and techniques is also of critical importance in order to offer a competitive advantage within the global information systems development environment. The lack of research explicitly focusing on the relationship between the implementation of Process Maturity Models and the use and effectiveness of Software Development Methodologies needs to be addressed as there is still a lot to be learned and potential advantages waiting to be discovered.

2.9 Previous research on Process Maturity Models and Software Development Methodologies integration

Research done on Process Maturity Models and Systems Development Methodology integration is few and far between. Table 2.4 indicates different search terms that were used during this research along with the results retrieved; the majority of applicable findings only have some relevance towards the subject at hand. Searches were conducted on a few of the major scientific databases. The total number of results found is displayed followed by the number of articles which were actually applicable to the subject. Each of the applicable results' titles and references is also indicated. Search criteria were limited to articles with matching; titles, abstracts, and keywords.

Table 2.4 – Search Results

Keywords:	<i>System* Develop* Method* AND process maturity integrat*</i>	<i>System* Develop* Method* AND process maturity</i>
ScienceDirect	4(2)	14(6)
	1. A CMMI appraisal support system based on fuzzy quantitative benchmarks model (Cheng <i>et al.</i> , 2011)	1. Software process maturity and certification (Bicego & Kuvaja, 1996)
	2. An encompassing life cycle centric survey of software inspection (Laitenberger & DeBaud, 2000)	2. The maturity of maturity model research: A systematic mapping study (Wendler, 2012)
		3. Case study: The effect of IS maturity on information systems strategic planning (Cerpa & Verner, 1998)
		4. A CMMI appraisal support system based on fuzzy quantitative benchmarks model (Cheng <i>et al.</i> , 2011) Repeat Article
		5. Producing reliable software: an experiment (Smidts <i>et al.</i> , 2002)
		6. Process models for service-based applications: A systematic literature review (Lane & Richardson, 2011)
Scopus	20(5)	59 (11)
	1. A CMMI appraisal support system based on fuzzy quantitative benchmarks model (Cheng <i>et al.</i> , 2011) Repeat Article	1. Identifying strategic management information systems planning parameters using case studies (Ang <i>et al.</i> , 1995) Repeat Article

	2. System development planning using readiness levels in a cost development minimization model (Magnaye <i>et al.</i> , 2010)	2. Software process improvement problems in twelve software companies: An empirical analysis (Beechan <i>et al.</i> , 2003)
	3. IQM3: Information quality management maturity model (Caballero <i>et al.</i> , 2008)	3. Software process maturity and certification (Bicego Kuvaja, 1996)
	4. An Evolutionary Software Management Maturity Model for Mauritius (Sukhoo <i>et al.</i> , 2007)	4. IQM3: Information quality management maturity model (Caballero <i>et al.</i> , 2008) Repeat Article
	5. Identifying strategic management information systems planning parameters using case studies (Ang <i>et al.</i> , 1995)	5. Supporting CMMI Level 2 SAM PA with non-technical features catalogues (Carvallo <i>et al.</i> , 2008)
		6. Case study: The effect of IS maturity on information systems strategic planning (Cerpa & Verner, 1998) Repeat Article
		7. On the limitations of software process assessment and the recognition of a required re-orientation for global process improvement (Gray & Smith, 1998)
		8. A maturity model and an evaluation system of software customer satisfaction: The case of software companies in Korea (Leem & Yoon, 2004)
		9. Story card Maturity Model (SMM): A process improvement framework for agile requirements engineering practices (Patel & Ramachandran, 2009)
		10. Evolutionary Software Project Management Maturity Model for Mauritius (Sukhoo <i>et al.</i> , 2007)

		Repeat Article
		11. Creating software process capability/maturity models (Wangenheim <i>et al.</i> , 2010)
EbscoHost	12(3)	18(6)
	1. Quantitative Framework for Managing Software Life Cycle (Stoica <i>et al.</i> , 2011)	1. Factors that Affect Software Systems Development Project Outcomes: A Survey of Research (McLeod & MacDonnel, 2011)
	2. Supporting CMMI Level 2 SAM PA with non-technical features catalogues (Carvallo <i>et al.</i> , 2008) Repeat Article	2. Quantitative Framework for Managing Software Life Cycle (Stoica <i>et al.</i> , 2011) Repeat Article
	3. Integrated Software Engineering and Knowledge Engineering Teaching Experiences (Dieste <i>et al.</i> , 2000)	3. Intangible benefits of CMM-based software process improvement (Hyde & Wilson, 2004)
		4. Capability Maturity Modeling SM at the SEI (Konrad <i>et al.</i> , 1996)
		5. QUALITY MANAGEMENT IN SYSTEMS DEVELOPMENT: AN ORGANIZATIONAL SYSTEM PERSPECTIVE (Ravichandran & Rai, 2000)
		6. Based on Quantification Software Quality Assessment Method (Yang & Zhang, 2009)
IEEE Xplore	7(2)	10 (2)
	1. A single model for process improvement (Ibrahim & Pyster, 2004)	1. A single model for process improvement (Ibrahim & Pyster, 2004) Repeat Article

	2. Unifying software engineering and systems engineering (Boehm, 2000)	2. Unifying software engineering and systems engineering (Boehm, 2000) Repeat Article
--	--	--

As can be seen from Table 2.4, not a lot of research has been done on the subject of investigating the relationship between Process Maturity Models and the use and effectiveness of Systems Development Methodologies. The majority of articles mentioned only apply the above mentioned concept to some extent; instead most of the articles found merely focus on Process Maturity Models as a technique which aids in increasing information systems development quality.

2.10 Chapter summary

In this chapter we reviewed the history of Systems Development Methodologies, its advantages as well as its criticisms. A framework for identifying and comparing SDMs was also provided and an example Systems Development Methodology for each of the identified categories was discussed. Following the framework discussions, we compared one SDM from each category, reviewing shared similarities and comparing differences. The ISO 9000 standards were also reviewed by examining its history along with its certification requirements; advantages of being certified in ISO 9000 were reviewed followed by critique against its implementation. The Capability Maturity Model Integration was also reviewed by examining its historical origins and use; reviewing its maturity levels and what is required for each appraisal level; and lastly advantages of being certified in CMMI as well as critique against its implementation. Finally CMMI and SDM integration was examined followed by previous research done on the subject. Chapter 3 focuses on the research design; which research paradigms were available; and the research approach applied in this study.

Chapter 3 - Research Design

3.1 Introduction

Research on the use and effectiveness of Systems Development Methodologies and Process Maturity Models can be based on a number of possible research approaches. Factors such as the research question, research goal, research target, theoretical orientation of the study as well as the research method can have an influence on the research approach.

As stated in Chapter 1 the research issue this dissertation posed was to “examine the relationship between Process Maturity Models and the use and effectiveness of Systems Development methodologies”. Chapter 2 examined Systems Development Methodologies as well as Process Maturity Models.

In this chapter different research approaches will be examined, the paradigm used in this study is discussed and justified. Following this, the research method, data collection method and data analysis techniques are discussed. The layout of this chapter is depicted in Figure 3.1 below:

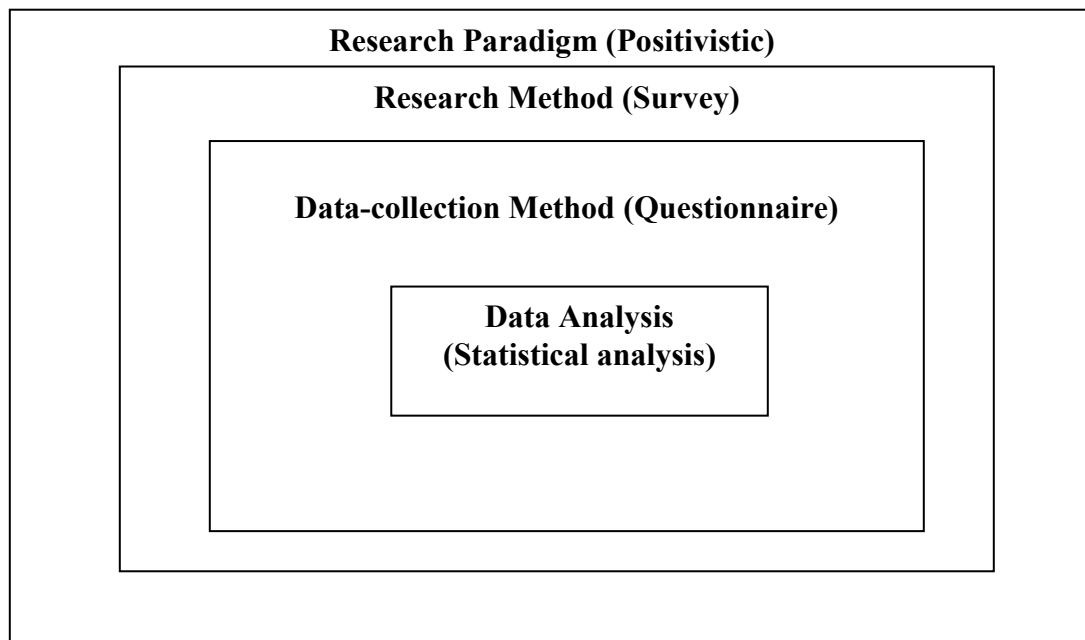


Figure 3.1 – Research design layout

3.2 Research Paradigms

Research paradigms are concerned with carrying out research within a specified area in a particular manner. To further understand this term we can examine the definition of the word 'paradigm'. Merriam-Webster (2011) defines it as follows: "a philosophical and theoretical framework of a scientific school or discipline within which theories, laws, and generalizations and the experiments performed in support of them are formulated". Accordingly it can be referred to as a set of shared assumptions or a way of thinking about some aspect of the world. Each philosophical paradigm has a different view of the nature of our world (*ontology*) and the means we go through to acquire knowledge in regards to it (*epistemology*). This in turn leads to research which is affected by the specific paradigm and is reflected by the research strategies used. Three main types of research paradigms exist, namely:

- Positivistic (scientific-positivism),
- Interpretive, and
- Critical social.

Each of these paradigms will be examined and the one most suited to the research being conducted will be implemented in order to aid in collecting data, analysing said data, and then being able to draw conclusions (where possible) to answer the question posed by the research, namely: to determine whether a relationship exists between Process Maturity Models and the use and effectiveness of Systems Development Methodologies.

In conducting research, Hair *et al.* (1995) suggest that studies which focus on seeking to confirm or test a certain hypothesis are referred to as confirmatory studies, whereas studies that define possible relationships in their most common form and then continue by using multivariate techniques to approximate a relationship(s) is referred to as exploratory studies. These studies (in the latter's case) allow the data and the method of investigation to define the nature of the relationships. The nature of the expected data from this study will be discussed later in this chapter, and the data as well as the analysis of the processed data is discussed in the next chapter, Chapter 4.

3.2.1 Scientific-Positivism

The term positivism (often regarded as “science”), was invented by the alleged founding father of sociology, Auguste Comte (1798-1857). The term is shorthand for Logical Positivism, or, more generally, to assign any approach that applies the scientific method to the study of human action. This defines positivism as the position that holds that facts and values are distinctive, and scientific knowledge consists only of facts.

The scientific-positivism paradigm has many defining characteristics. Firstly there is a focus on positive data, in other words, facts that can be identified and can survive attempts at falsification (Tribe, 2001:443). A scientific method is used to gather information, based on hypothesis formulation and testing against imperial evidence. The Falsification Principle is the core of the positivist paradigm. This principle entails that experience can show theories to be wrong, but never be able to prove them right. The underlying concept of the Falsification Principle is consequently the fact that the accuracy or correctness of theories can never be shown (Straub *et al.*, 2004).

3.2.2 Interpretive

This paradigm seeks to understand and identify the meaning behind the research results. The interpretive research paradigm treats the social world as a subject whereas the scientific-positivist treats the social world as an object. This helps to facilitate “the world to speak for itself”. Tribe argues that the interpretive research paradigm amplifies human actions and social constructs and as such they cannot be treated in the same way by researchers as natural objects (Tribe, 2001:445).

Furthermore, one should distinguish between interpretive research and a related term, qualitative research. Although interpretive research is usually qualitative in nature, it should be noted that qualitative research can be both interpretive and positive, depending on the philosophical assumptions of the researcher (Rowlands, 2005:81).

Schwandt (2007) reports that qualitative research can be the collective term for various techniques which seek to decode, describe, translate and by some means come to terms with the inherent implication(s), as opposed to measurement or rate of recurrence of certain

phenomena in the social world. In layman's terms this refers to the usage of text as opposed to numbers.

Diverging from qualitative research, interpretative research is more focused and specific in nature and is primarily defined in terms of epistemology. Interpretive research does not predefine any dependent or independent variables. Its aim is also not to test hypotheses, but rather to aim at fabricating an understanding of the social context regarding the phenomenon; including the method whereby the phenomenon influences and is influenced by the social context (Walsham, 1995).

3.2.3 Critical social

A number of ontological and epistemological positions can be defined as critical. These positions were developed by a variety of social theorists and social thinkers. One of the main supporters of the critical theory is the Frankfurt School (Horkheimer, Adorno, Marcuse, Habermas, and others). Other examples include the actor-network theory, the feminist theory, and Marxism (Howcroft & Trauth, 2004; Stahl & Brooke, 2008).

Research with the intention and motivation to change some social realities and promoting emancipation is referred to as 'critical' research. Critical research is characterized by the use of critical methodologies, the focus being on critical topics and the implementation of critical theories (Stahl, 2008).

3.3 Research paradigm used in this dissertation

The paradigm to be used in this dissertation is the scientific-positivist approach. Seeing that this research investigates a relationship between factors, the scientific paradigm will best suit these requirements. This chapter focuses on said paradigm, and the different advantages and disadvantages as well as methods used for collecting data are examined. The data collected will be used to justify the application of this specific paradigm for usage within this research. The application of principles to ensure research quality is also examined and discussed. Each principle will be listed and its application in this research justified.

3.3.1 The Scientific Research Paradigm

The scientific method is the oldest and most widely used paradigm of the three main paradigms. Since the term was coined in the early 18th century by Auguste Comte (1787-1857), the scientific method has been constantly reviewed and improved to evolve into a paradigm which many perceive as the only approach to deliver ‘proper’ research result (Schwandt, 2007).

The scientific-positivist approach uses a variety of verifiable scientific methods when gathering data. Brown (1977) summarizes that logical positivist, also known as the “empiricist” position that philosophically focuses on science as its objective and furthermore places an emphasis on methodical measurement and hypothesis testing. He also continues by stating that the scientific-positivist model is characterized mostly by its use of statistical methods and that any statement or proposal is only meaningful if it can be empirically verified.

3.3.2 The Scientific method

The scientific method entails two basic assumptions (Oates,2006:283):

- The world we live in has a form of order which is simply not random, and
- as a result it can be investigated objectively.

The first assumption entails that a certain situation will always deliver the exact same results, for example: an apple that falls from a tree will always fall downward, towards earth. This action will be the same no matter where the tree is. This is a result of the earth’s gravitational pull. Researchers are also trying to find patterns or laws in other areas, for example:

- Would a person’s chances of getting skin cancer increase with prolonged exposure to sunlight?
- An increase in global temperatures will cause the polar icecaps to melt.

The second assumption of positivism entails that the world with its regular laws and patterns can be investigated objectively. This means that research done isn’t dependent on the researcher and how the researcher personally experienced it. Any personal feelings should be set aside and researchers should be objective and rational.

The goal of the scientific and positivist paradigm is to find universal laws, patterns and regularities. This is mainly done through conducting experiments and analysing the results. Research should be done to disprove a theory rather than confirm it. This is because we can never prove that something will be true for all time. The hypothesis can be true for years, until a situation occurs or comes into existence that may disprove the whole theory. Therefore all theories contained and used should be regarded as the best knowledge we have of our current situation.

Three different basic techniques exist in the scientific method: reductionism, repeatability and refutation:

- **Reductionism:** entails the breaking down of concepts, theories, into smaller parts in order to be more easily studied.
- **Repeatability:** an experiment is repeated many times to see whether it consistently delivers the same result.
- **Refutation:** if an experiment cannot be repeated by other researchers and its reported results cannot be duplicated, the hypothesis is then refuted. This technique is also applicable when an experiment cannot create the same results in different circumstances.

The scientific method builds up knowledge through an iterative cycle:

1. Formulate a theory on an observation of the world.
2. Derive a hypothesis.
3. Objectively test the hypothesis.
4. Examine results.
5. Confirm or refute the hypothesis.
6. Accept, adjust or reject the theory.

3.3.2.1 *Characteristics of the positivistic research*

The positivistic research paradigm can be identified by various characteristics; these characteristics typically form the foundation of the said paradigm. These characteristics determine the type of research, research methods, objective etc. of the paradigm. A few of

the positivistic research paradigm's characteristics (Oates, 2006:286; Stahl, 2008; Brown, 1977) include:

- The world exists independently of humans.
- The researcher 'discovers' this world by making measurements and observations.
- The researcher is objective in his/her findings.
- Tests are conducted to confirm or refute the hypothesis.
- Data can be mathematically analysed to provide an objective means of analysing and observing results.
- Research looks for universal laws or patterns and facts that can be proven to be true regardless of the researcher or situation.

3.3.2.2 *Principles employed by the positivist research paradigm*

The principles of a research paradigm indicate whether the research findings are accurate and the whether the findings can be considered as valid. The quality of positivist research can be judged by the following principles (Popper, 1959; Durbin, 1968):

- Objective: was the research conducted free of the researcher's personal interests? Has the researcher influenced the results in any way? *This study was conducted purely independent of the researcher's personal interests and results were in no way affected by the researcher.*
- Reliable: were the methods/instruments used in measuring/obtaining the research results accurate and reliable? *The questionnaire used in this research was tested thoroughly and the research findings were obtained by using professional statistical analysis tools.*
- Internal validity: was the research well designed, so that the researchers examined the right things, or collected the data from the correct sources? *This questionnaire used in this research was thoroughly investigated before being sent out to respondents. Each question served a purpose and added to the value of the research. By employing a non-probabilistic purposive sample, the research ensured that the correct sources were used for its data collection.*
- External validity: can the research be generalized to different people, settings or times? Positivist research seeks high 'generalizability'. The research was designed in such a

manner as to best describe the population as a whole and thus can be applied across the entire field within information systems development.

3.3.3 Criticisms of the positivist research paradigm

Positivism is well suited to be used for studying natural world activities or aspects. However, it is less suited to be used in studying the social world, in other words: people, organizations, cultures, etc.

The following aspects can be considered as drawbacks from using the scientific-positivism approach:

- Breaking concepts into smaller parts to be studied is not always possible or effective.
- Repeating the research is not always possible.
- Generalizing is not always desirable.
- Different people view the world in different ways.

Despite these restrictions/disadvantages, the scientific-positivist approach offers the best possible methods and tools to conduct this research and deliver sensible valid results.

3.3.4 Research methods used in the positivistic paradigm

Various methods are available for collecting data when using the scientific-positivist research paradigm. These methods are better suited in specific environments or when certain factors influence the accessibility of the data 'sources'.

Two of the main research data collection methods in the scientific-positivism paradigm include:

- Surveys, and
- Experiments.

Experiments: Experiments are defined by Merriam-Webster as operations or procedures carried out under controlled circumstances in an attempt to discover an unknown effect or law, to test, confirm, or establish a hypothesis, or to illustrate a known law (Merriam-Webster, 2011).

Surveys: Surveys are methods used on a sample(s) to draw conclusions about the whole population. Merriam-Webster (2011) defines surveys as the query or solicitation of a person or group in order to collect data for the analysis of a group or area, often referred to as the population.

3.3.4.1 *Research method used in this study:*

For this research, surveys are used to collect data, as there is a specific target population from which quantitative data is needed in order to determine the use and effectiveness of Process Maturity Models as well as Systems Development Methodologies.

3.3.4.2 *Defining surveys:*

Surveys aim to obtain similar data from a large group of people (called a sample), in a standardized and systematic way. The data is investigated to see whether any patterns or similarities exist between the data, this is then used to draw conclusions about the larger population. According to Oates (2006:93) surveys are mostly associated with the positivism philosophical paradigm.

3.3.4.2.1 *The advantages/disadvantages of survey-based research*

For the researcher to be effective in terms of his or her research when using surveys as a research method, he or she needs to be aware of the advantages and disadvantages of using this particular research method. Some of the advantages/disadvantages can be listed as follows (Oates, 2006:104; Marshall, 2005:132; Jack & Clarke, 1998:176-179):

3.3.4.2.1.1 *Advantages of Surveys*

- Surveys provide a wide and inclusive coverage of people or events, so that the results are more likely to be representative of the wider population.
- They produce large quantities of data in a short amount of time and at a low cost compared to other strategies.
- They lend themselves to quantitative data analysis.
- The results can be replicated when collected from another sample or at a later time from the same sample.

- Electronic surveys or surveys via postal or web questionnaire, observations or documents are suited to people who don't have good inter-personal and communication skills.

3.3.4.2.1.2 Disadvantages of Surveys

- Surveys lack depth.
- They tend to focus on what can be counted and measured, and subjected to statistical analysis.
- They provide snapshots or particular points in time, rather than examining ongoing processes and change.
- They do not establish cause and effect.
- With postal, telephone, or Internet surveys, researchers cannot judge the accuracy or honesty of people's responses by observing their body language.

The above-mentioned criticisms/drawbacks of surveys may limit research results (depth) to some extent, but these limitations are greatly outweighed by the advantages which are offered in studies such as this. For this research surveys were sent out across South Africa. This process started in October 2010 and continued until July 2011. A list of possible respondents was obtained by conducting internet searches for software development companies within South Africa. These companies were contacted telephonically or alternatively via email whenever no contact number could be found. Within each responding company a contact person was identified through which the questionnaire would be distributed. The contact person mentioned was then contacted on a weekly basis in order remind them of responding to the questionnaire. After these contacts were exhausted and no further replies were being received, an alternative solution had to be found. Potential companies were contacted in person and a 'drop-off' approach was followed. The latter technique did prove to offer better response rates. After the required number of completed questionnaires had been returned, the data was analysed and summarised.

3.3.5 Survey data gathering methods

In order to collect data which can be used for analyses, certain techniques can be used to effectively represent the population. When using the survey data collection technique the following methods offer the opportunity to collect the required data:

- Interviews,
- Observations,
- Documents, and
- Questionnaires.

It should be noted that some of these methods are better suited in specific environments or cases and concomitant with the researcher's ideals and/or preferences, the chosen method(s) can have an influence on the accuracy or profundity of the results.

3.3.6 Planning and designing a survey

A total of six important activities are pertinent when planning and designing a questionnaire (Oates, 2006:94; Brace, 2008:35-44; Oppenheim, 2000:49-82). These activities are of critical importance, as a poorly designed questionnaire will affect the data gathered. Respondents are prone to neglect or ignore a poorly designed questionnaire, which may result in inaccurate or incomplete results. The six activities can be listed as follows:

1. Data requirements,
2. Data generation method,
3. Sampling frame,
4. Sampling technique,
5. Response rate and non-responses, and
6. Sample size.

These activities help in organizing and planning a survey. All these tasks play a critical role in the end result, and as such it is of great importance that all the activities be well-planned and correctly executed.

Brief descriptions of these tasks are presented next, as well as their application in this study:

3.3.6.1 *Data requirements*

A decision has to be made in terms of what type of data one wants to generate. The data can either be directly or indirectly related to the topic at hand. Normally only one chance exists to gather data from respondents, and so it is of great importance that all the required data is gathered on the first attempt. The data gathered in this research had to be quantitative in nature in order to summarise and compare the information systems development processes used by each individual/organisation. In this study various aspects of this development process had to be captured; a few of these include: organisation size, SDM usage intensity, historical data, etc.

3.3.6.2 *Data-generation method*

Data-generation methods are the methods used to retrieve data from sample(s). This retrieval method can include questionnaires, interviews, documents, and/or observations. Consequently a decision has to be made on which generation method will be the most effective in terms of your research topic. This research required a standardised set of information from a large audience and thus a questionnaire was chosen as the most applicable solution.

3.3.6.3 *Sampling frame*

A sampling frame is a group or collection of a whole population of people (or events) that could be included in the survey, from which a sample will be chosen. In this study the sampling frame consisted out of various Information Technology professionals, ranging from developers to managers, to business analysts. The sampling frame stretched to various business areas in order to obtain the best possible real-world representation, ranging from finance, to telecommunications, to straight out software development. The sampling technique applied is reviewed in 3.4.4.4.

3.3.6.4 *Sampling technique*

After obtaining a sampling frame, a sampling technique has to be chosen. Two types of sampling techniques exist, namely probability and non-probability sampling. Probability sampling refers to the sample that has been chosen because the researcher believes that the

chance is high that the population will be represented in the sample. Non-probability means that the researcher is unsure whether the sample will be an accurate representation of the population.

4.1 – Probability sampling techniques:

- Random sampling
- Systematic sampling
- Stratified sampling
- Cluster sampling

4.2 – Non-probability sampling techniques:

- Purposive sampling
- Snowball sampling
- Self-selection sampling
- Convenience sampling

In this study the purposive non-probabilistic sampling technique was used. A set of chosen companies were approached to complete the questionnaire. The companies approached during this research all specialised in software development within South Africa.

3.3.7 Response rate and non-responses:

Respondents are known to have a tendency to ignore questionnaires. A strategy is needed to try and increase the number of responses. This can vary from some sort of incentive or another form of motivation. In this study respondents were motivated to complete the questionnaire in order to determine their own level of Information Systems Development and to introduce respondents to other techniques which can aid their development processes. To further inspire respondents, especially bigger organisations, to complete the questionnaire; a picnic hamper was awarded to each organisation which returned six or more completed questionnaires.

3.3.8 Sample size:

A decision has to be made regarding the size of the sample, taking into account the expected response rate of participants. In order to accurately analyse the population according to the sample's responses, there needs to be sufficient data available. A total of 485 companies were contacted to complete the questionnaire. Only 125 out of the 485 responded, which leaves a response rate of just 25%. This is as to be expected from a questionnaire based survey.

3.3.9 Data generation methods

Data generation methods are the means by which one produces empirical data or evidence. The data can be produced in one of two forms: quantitative or qualitative. Quantitative data is numeric data, whereas qualitative data is all other types of data such as sound, images or words. Oates (2006:36) and Bell (1987) focused on four data-generation methods. These four data-generation methods can be defined as follows:

- **Interviews:** A conversation between people where, at least at the beginning of the interview if not all the way through, the researcher controls both the proceedings and the agenda, and will ask most of the questions. Interviews can either be one-on-one or group interviews.
- **Observations:** Watching people and observing their actions, rather than reporting what they do. This mostly involves the sense of sight, but might also involve tasting, touching, smelling or hearing.
- **Questionnaires:** A pre-defined set of questions arranged in a certain order. Respondents are asked to answer the questions, mostly via multiple-choice options.
- **Documents:** Documents that already existed prior to the research and documents that were created exclusively to be used for research purposes. Can include diagrams, electronic bulletin boards, videos, etc.

3.3.10 Data-generation method used in this research:

In this research a questionnaire was used as a data generation method due to the fact that the data to be collected was mainly quantitative. Oates (2006:229) and Marshall (2005:132) list the advantages and disadvantages of questionnaires as follows:

3.3.10.1 Advantages of Questionnaires

- Questions are more of an economical option compared to other data-generation methods.
- Pre-defined answers make it easier for respondents to complete and easier for researchers to analyse.
- Few geographical limitations exist where questionnaires cannot be used, seeing that many different distribution methods are available.
- Self-administration questionnaires require little to no unusual skills of the researcher.

3.3.10.2 Disadvantages of Questionnaires

- Pre-defined answers can cause frustration, which will in turn cause some questions to go unanswered which results in weaker results.
- A researcher cannot control the veracity of answers.
- A researcher cannot correct misunderstandings, probe for more details, or offer explanations for help.
- Self-administered questionnaires are unsuitable for those with visual handicaps or poor literacy skills.

Despite the few disadvantages of questionnaires the benefits of using questionnaires greatly outweigh its criticisms when applied in this research.

3.3.11 Application of data-generation method

The questionnaire was designed to systematically guide the respondent through the process of determining whether their Information Systems Development standards are effective and how they can further improve this process. The questionnaire used in this research aimed to firstly determine the size of the respondents' organisation (business environment), years in existence, etc. This aided in determining where these organisations are in terms of their Information Systems Development maturity, seeing as bigger/older organisations tend to adapt some sort of development standard after a few years or when reaching a certain size. The questionnaire continues by examining project success rates and the use of standardised development methods. Lastly the questionnaire attempts to determine whether the respondent's organisation is certified in one of the Process Maturity Models or at the very

least whether some of the required practices are already being applied in their everyday development processes.

Table 3.2 summarises the data/objective measured by each variable. The applicable variable is provided, along with the number items which measure the applicable variable.

Table 3.2 – Information captured by questionnaire

Variable	Questionnaire	Number of items measured
<i>Background information</i>		
Country	✓	1
Organization year in existence	✓	1
Size of organization	✓	1
Organizational sector	✓	1
Skills level	✓	1
Role within organization	✓	1
Software procurement	✓	1
Information System Development department size	✓	1
<i>Systems development methodology</i>		
Reasons for not using SDMs	✓	Respondent specific 13 Items measured on a 5-point Likert scale
Methodology used - Type - Intensity of use	✓	Respondent specific Intensity was measured on a 5-point Likert scale
Period of use	✓	1
Horizontal use of SDM	✓	2 (Huisman, 2000)
Strictness of use	✓	1
<i>Process Maturity Models</i>		
PMM Usage -Type	✓	2 Items with Yes/No/I don't know options Specify addition PMMs

Reasons for PMMs use	✓	1
Reasons for not using PMMs		1
Perceived CMMI Maturity	✓	38 Items
<i>Project management methodology</i>		
Methodology used - Type - Intensity of use	✓	Respondent specific Intensity was measured on a 5-point Likert scale
Period of use	✓	1
Horizontal use	✓	2
Strictness of use	✓	1
<i>Project Outcome</i>	✓	1
Size of project	✓	1
Product outcome	✓	1
Product success	✓	11 items measured on a 5-point Likert scale (Huisman, 2000)
Process success	✓	9 items measured on a 5-point Likert scale (Huisman, 2000)

A trial version of the questionnaire was sent out to representatives from each applicable business area identified in the sample frame. The questionnaire was adjusted using the feedback obtained from the said trial and the final version was assembled. The reviewed questionnaire was then sent out to the sample frame. As discussed in 3.4.6 the response rate was a meagre 25%, which was as expected seeing as the minimum time required to complete the questionnaire was estimated as half an hour or more.

3.3.12 Data analysis

The data received should be captured and analysed, but firstly it should be determined what type of data was received.

Two types of data exist, namely:

- Quantitative, and
- Qualitative

The questionnaire used in this research obtained data which was quantitative in nature. In turn this data was then analysed using statistical inference. Some of the techniques used to analyse the data include:

- Weighted average (Grossman *et al*, 2006)
- Standard deviation (Saeed, 2000)
- Factor analysis (Child, 2006)
- Regression analysis (Hardle, 1990)
- Cluster analysis (Mrina, 2007)
 - Tree cluster
- Reliability analysis (Cortina, 1993)
 - Cronbach Alpha (Allen and Yen, 2001)

3.4 Chapter Summary

In this chapter we examined the different research paradigms available in order to conduct research. We started off by examining the Scientific-Positivistic paradigm, thereafter the Interpretive paradigm, and lastly the Critical social paradigm. Each discussion highlighted what makes that specific paradigm unique, and key processes were identified. After these discussions we reviewed the paradigm chosen for this research; The Scientific-Positivism paradigm. This review discusses the Scientific-Positivistic paradigm in further detail. Characteristics, principles employed by the said paradigm, as well as criticisms against the paradigm were reviewed.

The different research methods available were examined and the chosen method, surveys, was examined in further detail by also examining the advantages and disadvantages of its use. The discussion continued by identifying the key attributes required by the survey. Following these discussions we identified some advantages and disadvantages of surveys as a research method during this study.

Lastly we discussed the application of the chosen data-generation method as well as well as reviewing how the data was analysed in this study.

Chapter 4 will focus on the research results, indicating the use and effectiveness of systems development methodologies and process maturity models in South Africa.

Chapter 4 – Research Results

4.1 Introduction

This chapter reports on the current use of Systems Development Methodologies in South Africa, as well as Process Maturity Models, in particular the Capability Maturity Model Integration as well as ISO 9000-3. This chapter starts by examining the participating IS department as well as the environment from which the respondents completed the questionnaire. This is followed by examining the individual respondents to form an understanding of their SDM knowledge and use, as well as determining the organization's perceived CMMI level. In the last section of this chapter we present a description of SDM use and process maturity model certification within South Africa, as well as examine the relationship between Process Maturity Models and the use and effectiveness of Systems Development Methodologies.

4.2 Development Environment

In order to describe the development environment within IS departments, we gathered information regarding the business area of the respondent's organization, how long the organization has been in operation, the skills levels of the ISD employees, the size of the organization, the size of the ISD, the respondent's role within the organization and how/whether they procure software. Furthermore we examined the size of the respondent's last project, if any, as well as the project's outcome. In all instances, except the question on software procurement, the respondent was provided with a list of possible answers from which only one option could be chosen. The question on software procurement had a list of possible methods-used from which the respondents could choose more than one appropriate option.

4.2.1 Business area

In order to be able to determine each respondent's organization's involvement in Information System Development (ISD), we needed to determine their core business area. The following options were provided: Agriculture; Catering; accommodation and other trade; Construction; Education; Finance/Banking/Insurance; Manufacturing; Retail; motor repair services; Software

house/Software consulting; Wholesale trade & commercial agents; Transport; storage and communication; Other. The descriptive statistics for the Business area is summarized in Figure 4.1. The following results were obtained: Software house/ Software consulting (43%), Transportation, storage and communication (23%), Finance/Banking/Insurance (20%), Manufacturing (6%), Retail, motor repair services (4%), and Education (1%). These results are satisfactory and Software house/ Software consulting dominates the results. This confirms the modern day practice of outsourcing software development rather than having a dedicated team for this purpose.

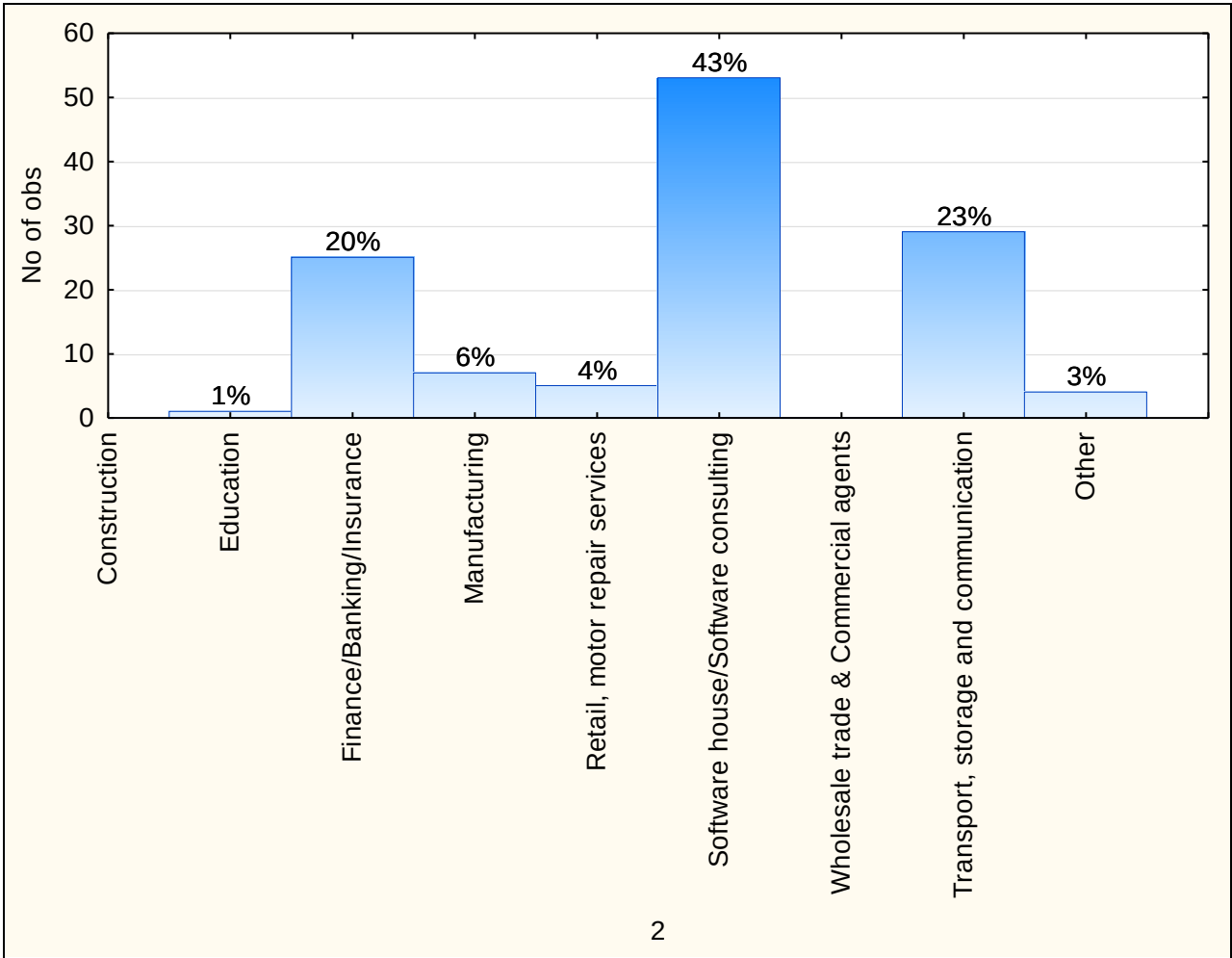


Figure 4.1 – Business Area

4.2.2 Role within the organization

Each respondent was also asked what his/her role within the organization was. The following options were provided: Owner, Executive Director, Manager, Project Leader, Developer, Tester, Business Analyst, DBA, other. The descriptive statistics for the respondent’s role within the organization is summarized in Figure 4.2. The results showed that 50% of respondents are/were Developers; this was followed by Project Leader (20%), Manager (9%), Executive Director as well as Tester with 6%, Business Analyst (5%), Owner (3%), Database Administrator (1%) and Other (1%). This conforms to the standards normally associated in an IT department – more developers than managers.

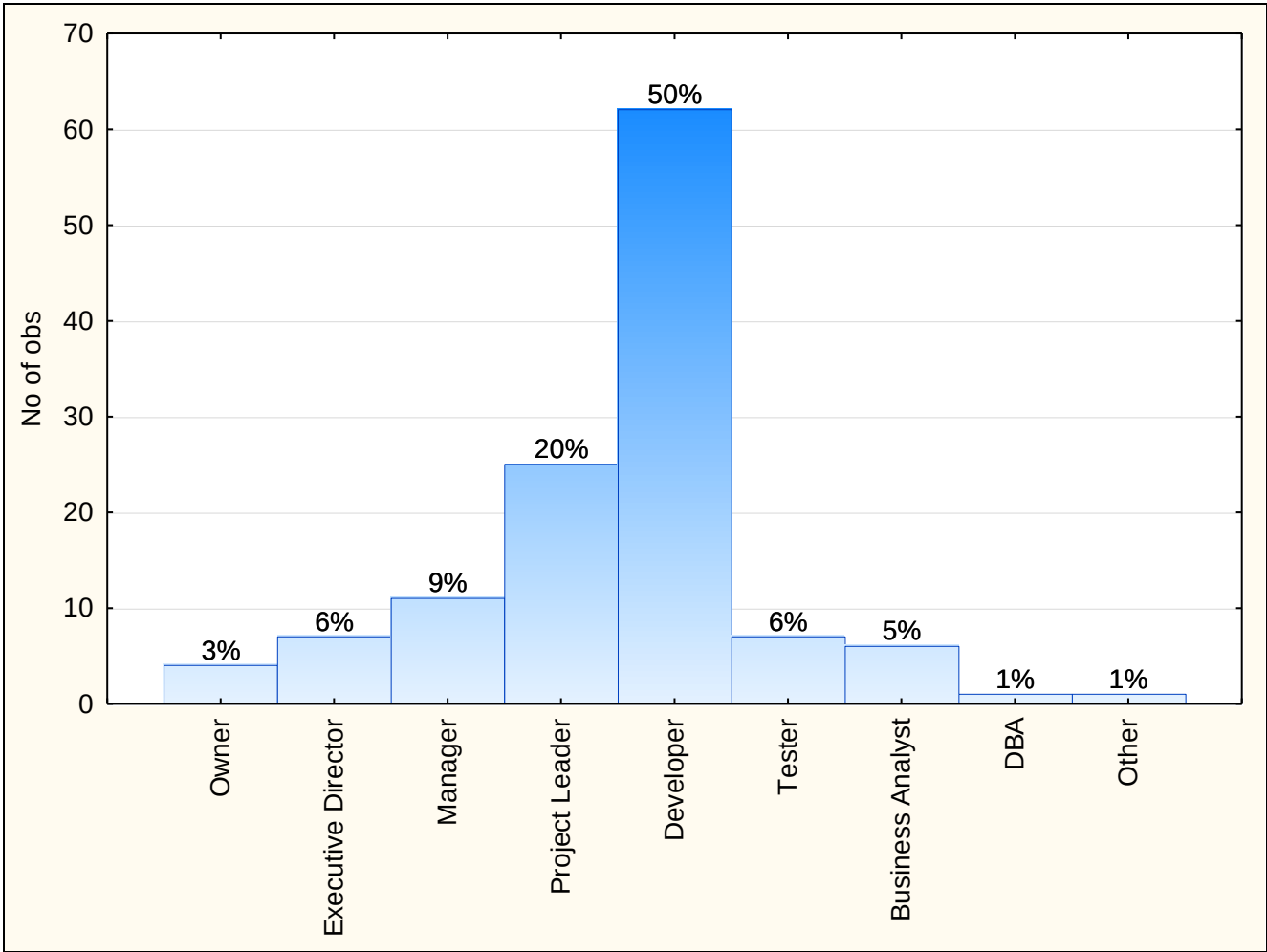


Figure 4.2 – Role within the Business

4.2.3 Organization’s years in operation

In order to determine how long each respondent’s organization has been in operation, the following options were provided: Less than 1 year, 1-3 years, 3-6 years, 6-9 years, more than 9

years. The descriptive statistics of the organization’s years in operation is provided in Figure 4.3. A total of 75% of respondents’ organizations has been operational for more than 9 years, followed by 3-6 years (15%), 1-3 years (7%), 6-9 years (3%), and less than 1 year (1%).

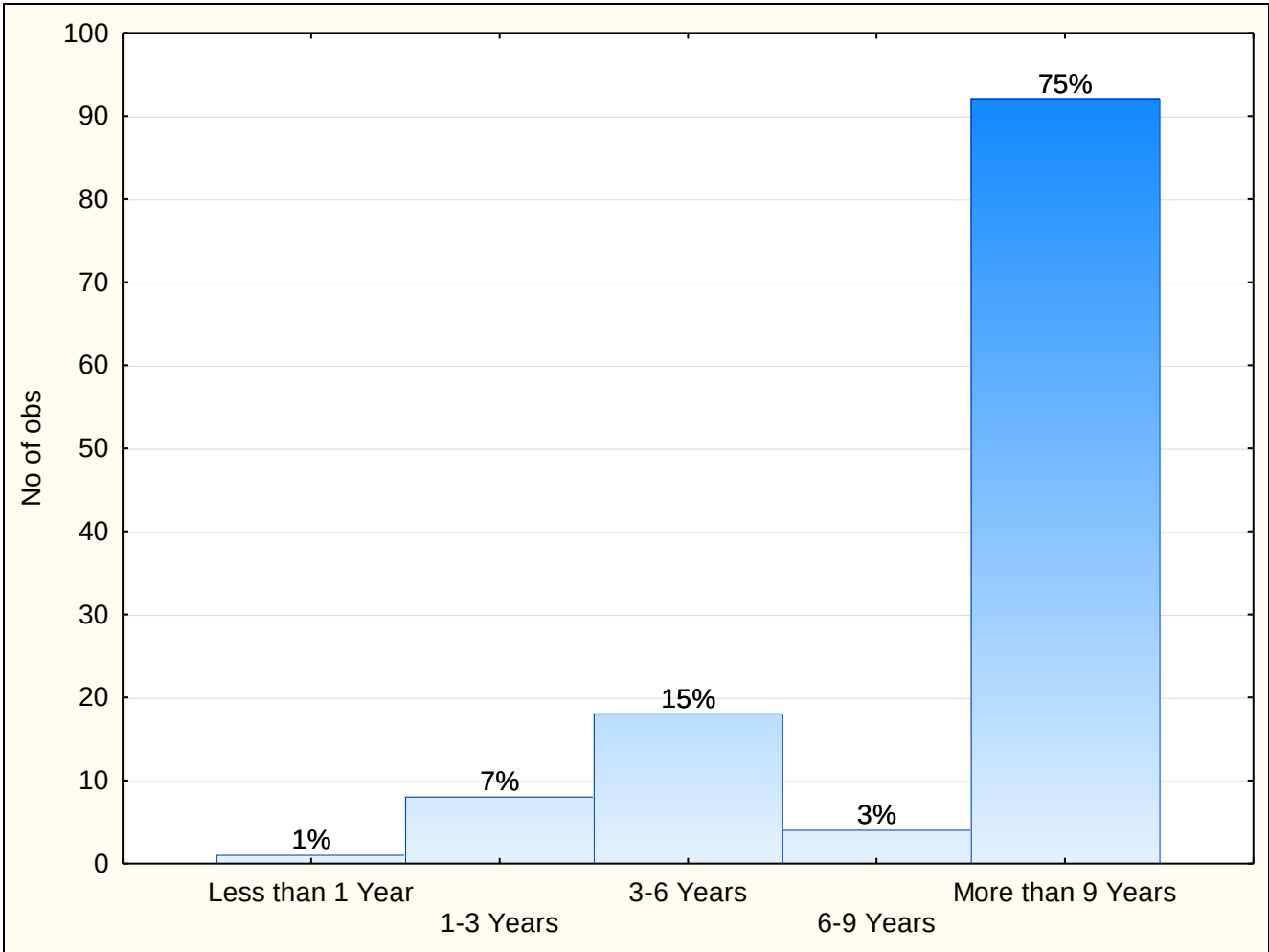


Figure 4.3 – Organization: Years Operational

4.2.4 Size of the organization

In order to determine the respondents’ organization size (in terms of total number of employees), respondents were provided with the following options: Micro (1-5), Small (6-50), Medium (51-100), Large (101 or more). This classification is in accordance to the South African Small Business Act 102 of 1996. The descriptive statistics for the organization’s size is provided in Figure 4.4. The results show that 65% of respondents’ organizations have more than 100 employees total; this was followed by Small (6-50) with 24%, Medium (51-100) with 7% and lastly Micro (1-5) with 4%.

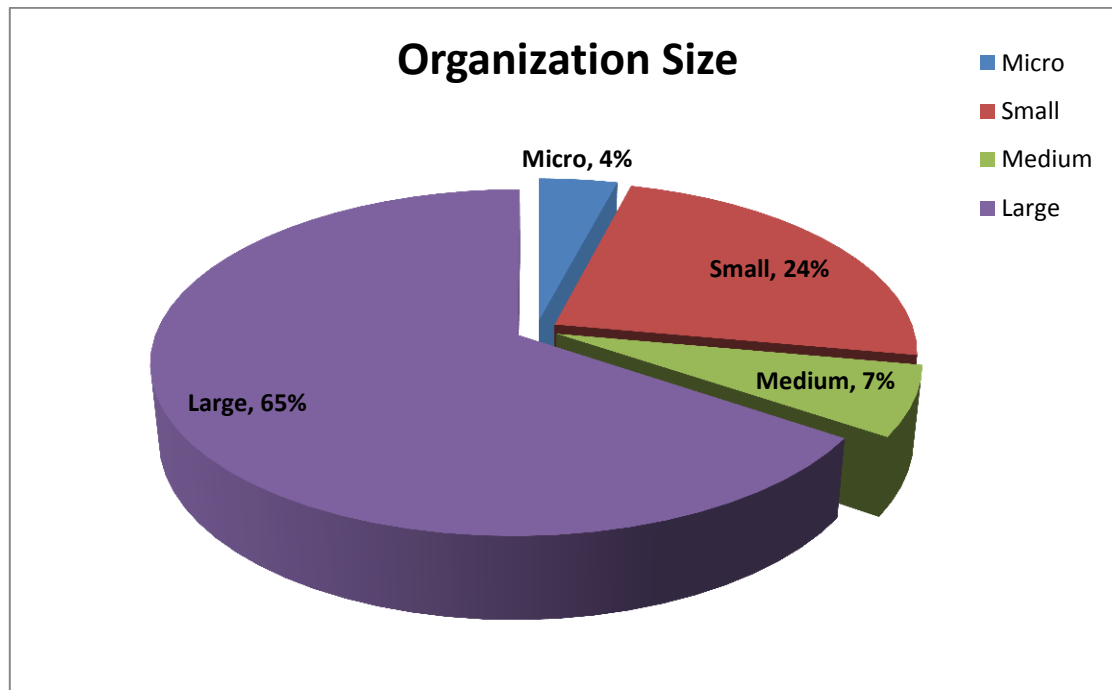


Figure 4.4 – Organization Size

4.2.5 Size of the ISD department

The size of the ISD department gives a more accurate view on how many employees are actually involved in the ISD process and was measured by the following options: 0 (none), 1-5, 6-10, 11-20, 21-50, More than 50. The results show that 30% of the respondents' ISD departments have more than 50 people employed within the ISD department; this is followed by 11-20 (29%), 21-50 (20%), 1-5 (12%), and 6-10 (9%). The descriptive statistics for the ISD department size can be viewed in Figure 4.5. The above-mentioned distribution can be considered as an accurate real-world representation of IS department sizes.

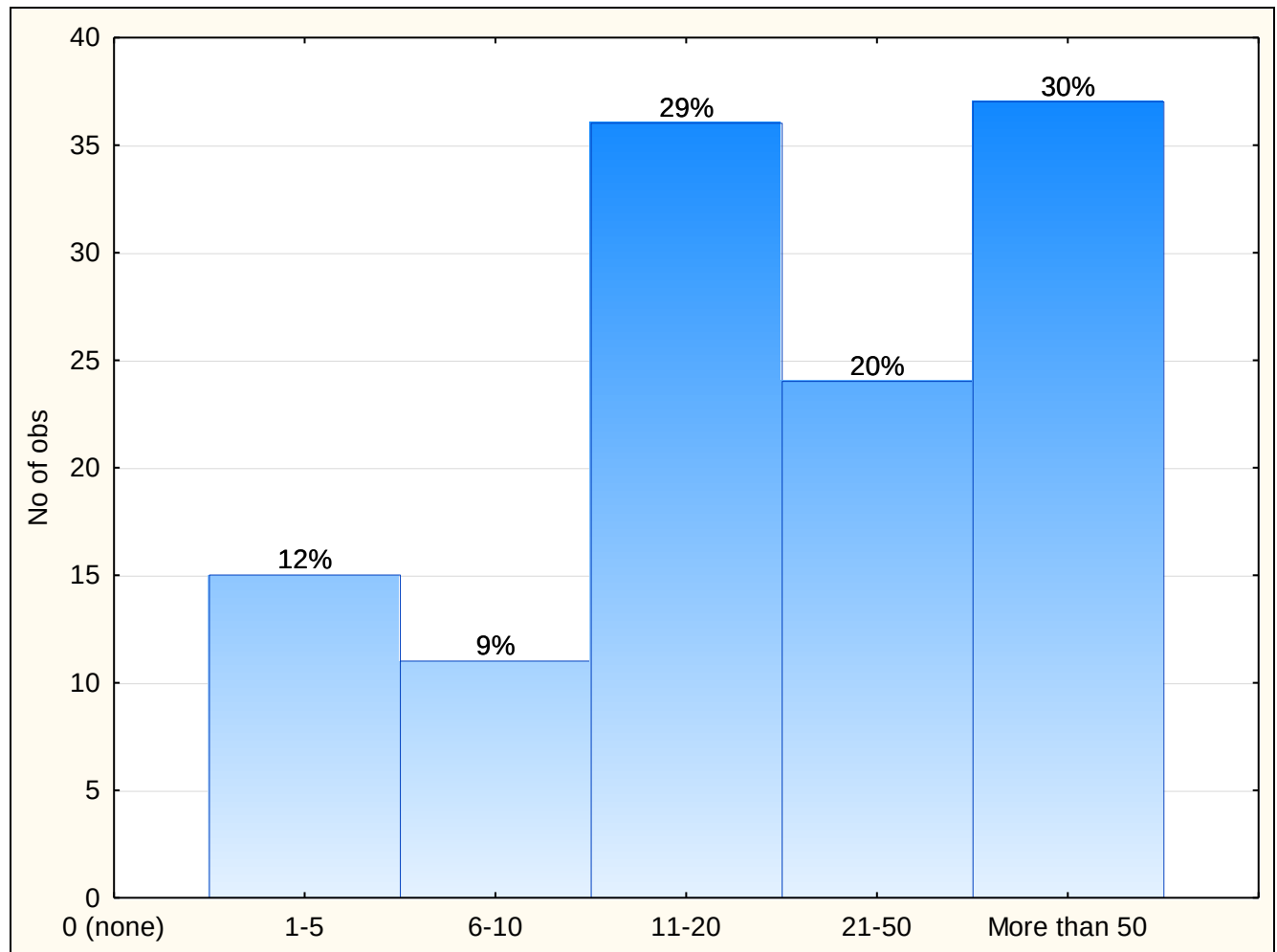


Figure 4.5 – Total Number of Employees within the ISD Department

4.2.6 Information System Development (ISD) Skill Level

The list of possible options to determine the general skill of employees involved in ISD was as follows: No information system development skills, Limited skills, Fairly skilled, Well skilled, Experts. The descriptive statistics of the general skill of employees involved in ISD is provided in Figure 4.6. Results show that 56% of respondents viewed the general skill of their ISD employees as Well skilled, this was followed by: Experts (27%), Fairly skilled (14%), and Limited skills (2%). This leads to the conclusion that respondents regard themselves as highly skilled individuals.

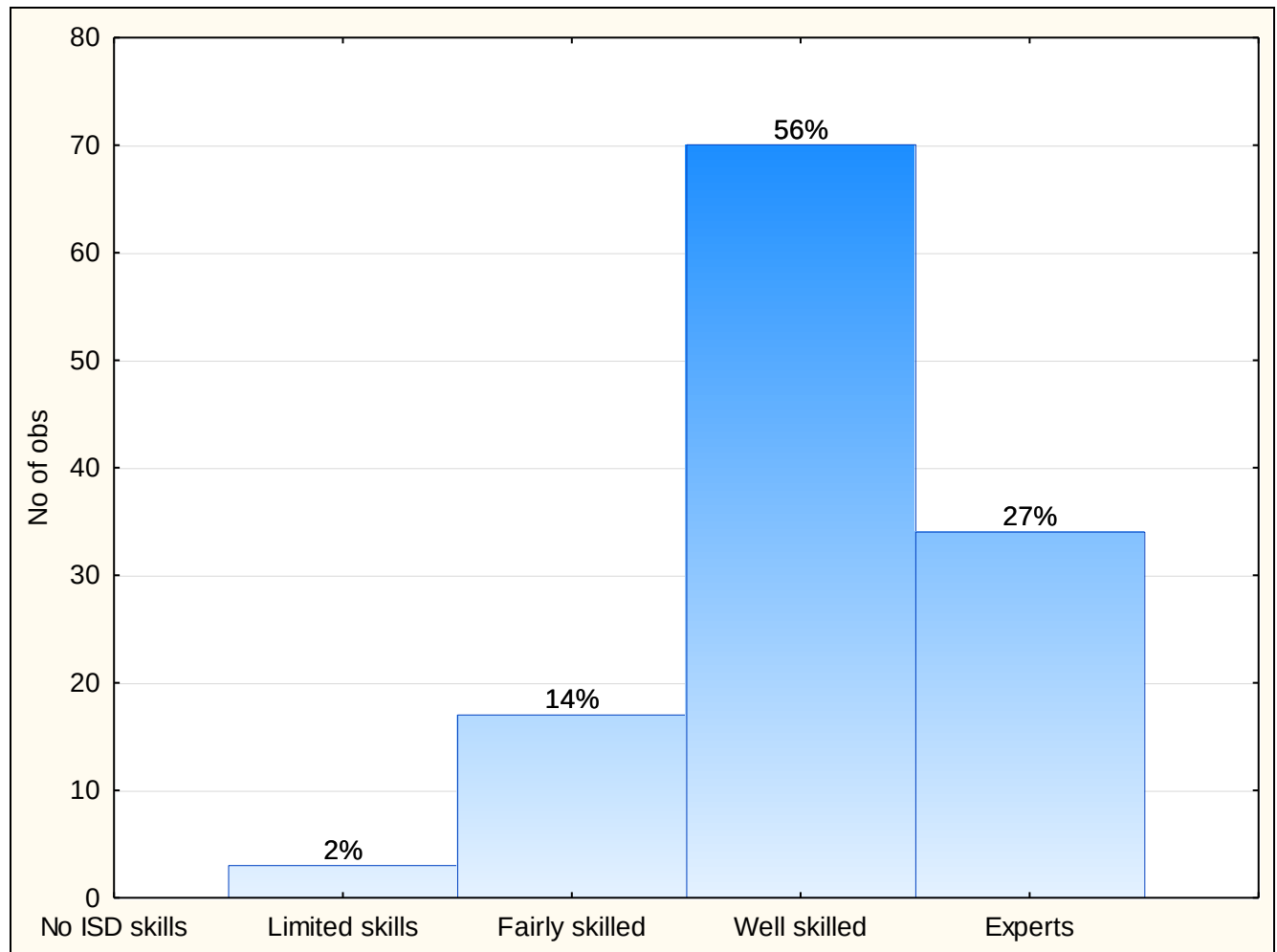


Figure 4.6 – ISD Employee Skill Level

4.2.7 Software Procurement

Software procurement had the following options available (more than one could be chosen):

- We do not procure any software,
- In-house development,
- Off-shelf (no customization),
- Off-shelf (with customization),
- Out-sourcing,
- Other.

Respondents could choose more than one option, and the results were as follows: In-house development was the most widely used with a total of 66%, this was followed by Off-shelf (with customization) (56%), Off-shelf (no customization) (40%), Out-sourcing (38%), and We do not procure any software (3%). The average number of procurement techniques used was two

(2) per respondent. The majority of software house/ software consulting respondents answered with “In-house development” and “Off-shelf (with customization)”, whereas companies which do not have their own ISD department mainly answered with “Off-shelf (no customization)” and “Out-sourcing”. The descriptive statistics for Software Procurement is summarized in Figure 4.7.

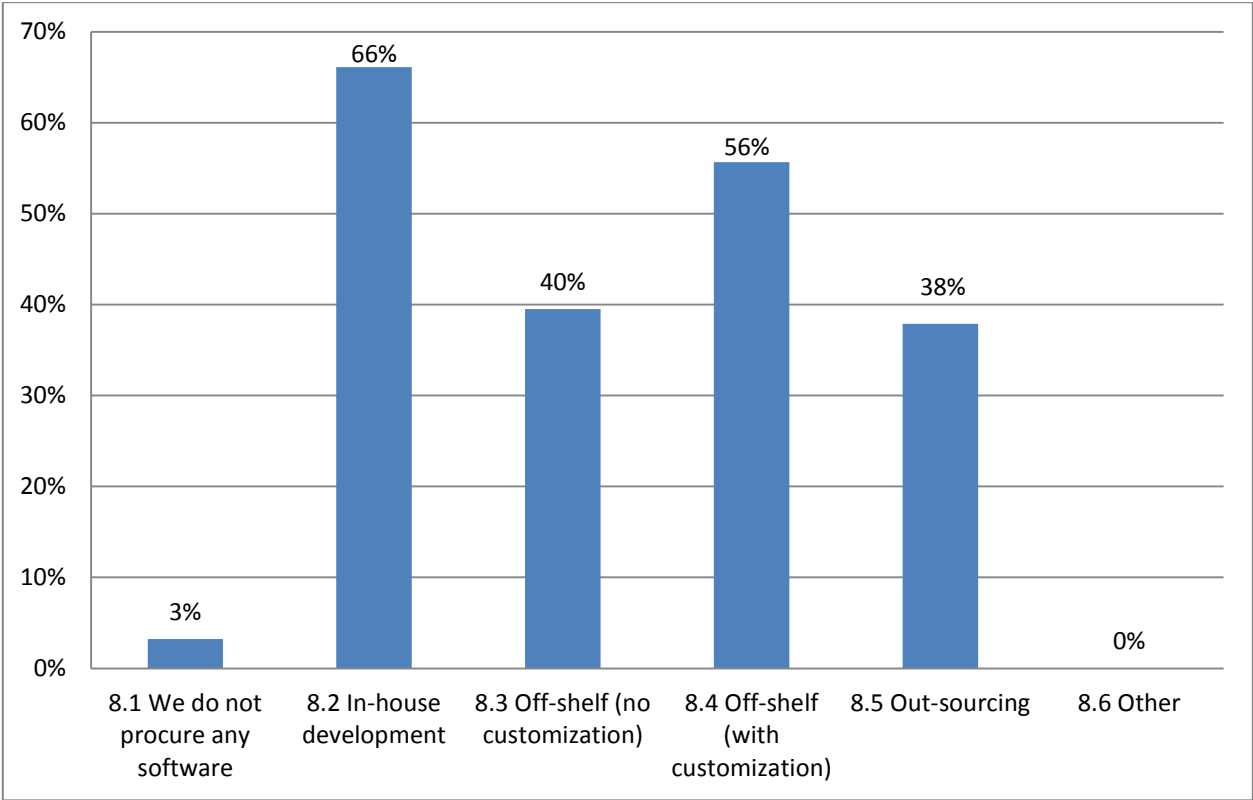


Figure 4.7 – Software Procurement

4.2.8 Project outcome

Project outcome (in terms of duration, budget, and other factors which influence success) was measured by the following options:

- I have not participated in any ISD projects,
- The project was cancelled/terminated before completion,
- The project was completed but not implemented,
- The project was completed and implemented but not in use anymore,
- The project was completed, implemented and is still in use.

The descriptive statistics for the project outcome is summarized in Figure 4.8. A total of 89% of the respondents answered with “The project was completed, implemented and is still in use”. This was followed by “I have not partaken in any information system development projects” (4%), “The project was cancelled/terminated before completion” (3%), “The project was completed but not implemented” (2%), and “The project was completed and implemented but is not in use any more”.

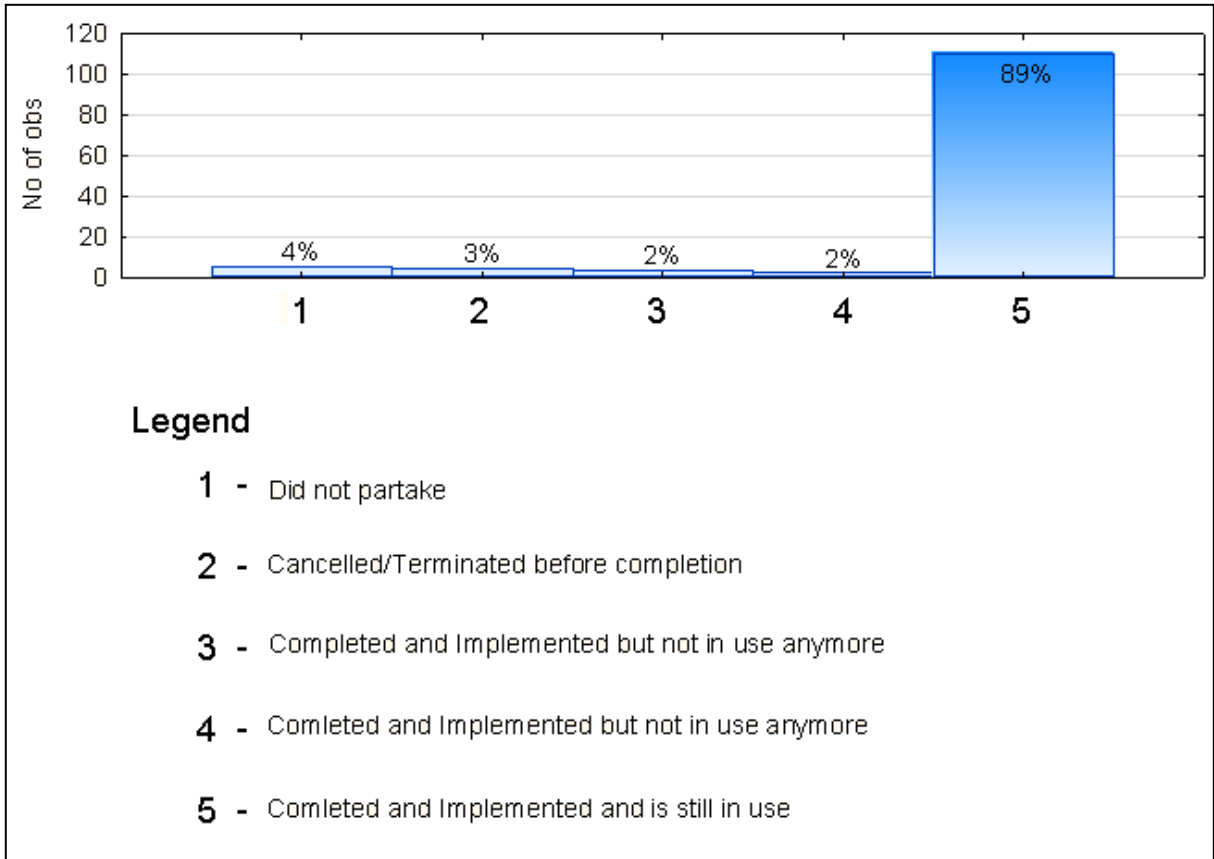


Figure 4.8 – Project Outcome

4.2.9 Project size

The size of the last ISD the respondents had been involved in could be answered by any of the following options: Small, Medium, Large, Very Large, and “I have not partaken in any ISD projects”. The descriptive statistics is summarized in Figure 4.9. The results were as follows: Large (41%), Medium (34%), Very Large (18%), Small (6%), and “I have not partaken in any ISD projects” with 2%. These results indicate a good representation of what is expected in the business world.

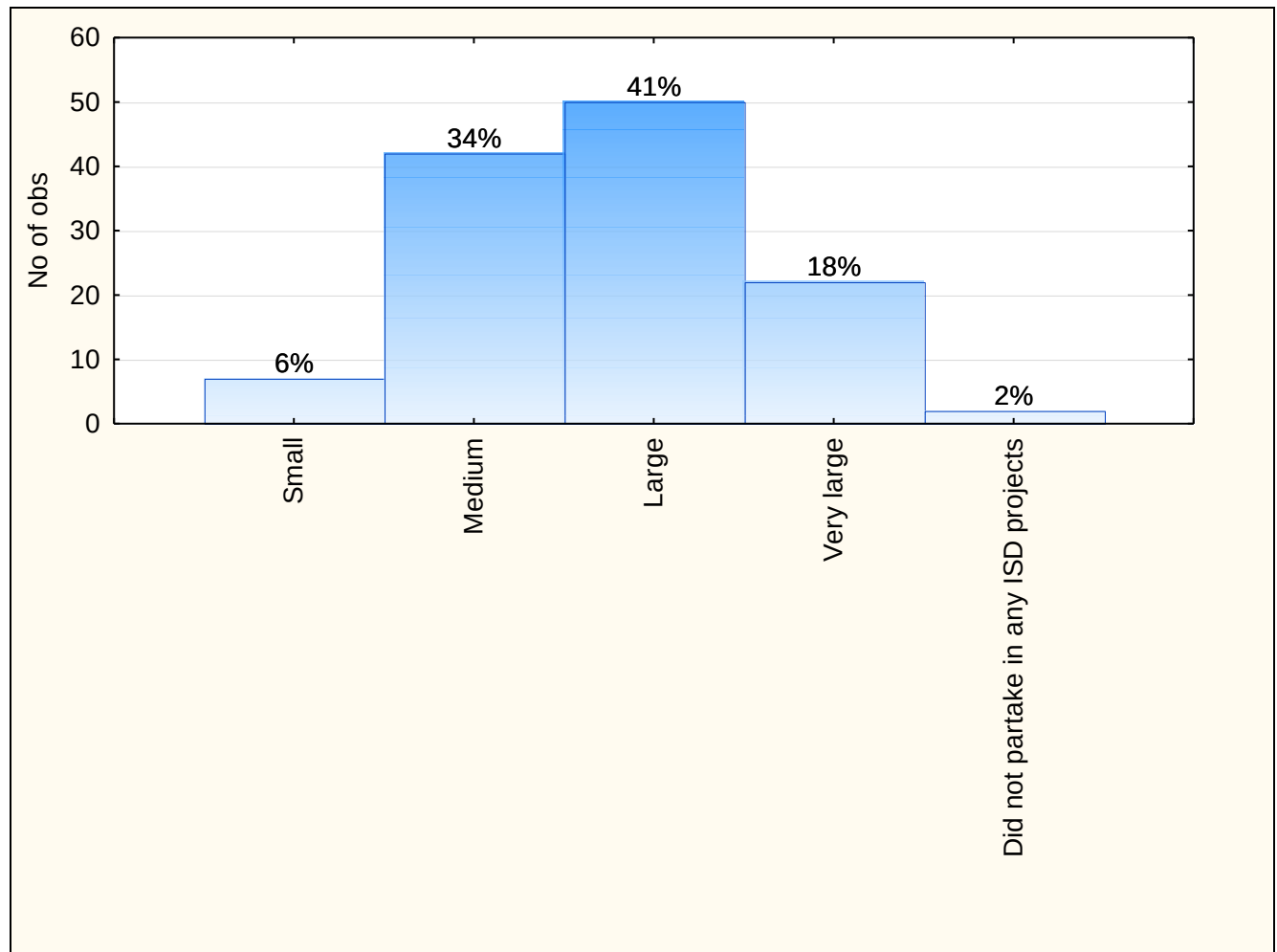


Figure 4.9 – Size Of The Last ISD Project

4.2.10 Development Environment Summary

The development environment proved to be versatile; ranging from business areas such as Finance, to Software Development/Consulting, and even Transport or communication. The majority of these organisations have been in operation for at least nine years and have more than 100 employees in total, where twenty or more of these employees are situated in their ISD department. This background information helped to determine how evolved a respondent's organisation was in terms of overall maturity as well as Information Systems Development maturity.

4.3 SDM Usage and Process Maturity certification

In this section we examine the different SDMs which the respondents use, as well as their motivation for using them. We continue by summarizing the reasons why some respondents found SDM usage unnecessary.

4.3.1 SDM usage

Respondents were asked to fill in the name(s) of the SDM(s) which their organization used, as well as the intensity of usage on a scale from 1-5(1 being used very infrequently and 5 being used very often). Out of a total of 125 respondents, 92 indicated that they do use a systems development methodology (73.6 %), from this a total of 35 (28%) of the respondents indicated that they implement two or more systems development methodologies. To measure the usage of System Development Methodologies, we focus on:

SDM usage: - Does the respondent's organisation use SDMs?

- Reasons for not using SDMs.

SDM Type used: - What types of SDMs are in use?

SDM historic usage: For how long have SDMs been in use in the respondent's organization.

Horizontal use: - What proportion of projects are developed in the respondent's IS department by using SDM knowledge?

- What proportion of people in the respondent's IS department apply SDM knowledge regularly?

Vertical use: - In relation to the number of projects being developed.

- With what intensity is SDMs used to help in the development process?

SDM usage strictness:- How does the respondent's ISD department use SDMs?

4.3.2 Reasons for not using an SDM

Respondents which were not using SDMs were asked to choose from the following reasons, summarized in Table 4.1 for not using an SDM:

Table 4.1.Reasons for SDM non-use

Reason for non-use:	Average:
The current system development practice in our IS department is	3.38

adequate	
The benefits of systems development methodologies use are long-term, whereas cost are incurred short term	3.21
The experience of the developers in our IS department reduces the need for systems development methodologies	2.78
In our IS department there are no clear objectives for adopting systems development methodologies	2.68
In our IS department there is a lack of management support for the use of systems development methodologies	2.55
The learning curve for systems development methodologies is very long	2.55
The financial investment in systems development methodologies is too large	2.52
In our IS department there is a lot of uncertainty over the benefits of adopting systems development methodologies.	2.51
Our IS department lacks a suitable environment to support systems development methodologies	2.49
The profile of development projects in our IS department doesn't require the use of system development methodologies	2.45
There is a lack of experienced staff in our IS department who can effectively use system development methodologies	2.44
New systems developed with systems development methodologies are not compatible with legacy systems	2.31
Systems development methodologies are too complex or hard to use	1.96

Respondents were asked to mark the most appropriate value at each question. Possible answers varied from 1 (one) to 5 (five), whereas 1 (one) was totally disagree and 5(five) totally agree. As seen from the table above Table 4.1 the two top reasons for not using an SDM are as follows:

- 1 - (3) The current system development practice in our IS department is adequate (average of 3.38).
- 2 - (5) The benefits of systems development methodologies use are long-term, whereas cost are incurred short term (average of 3.21).

As we can see, the majority of respondents felt that their current method of developing an Information System was adequate to successfully complete the project. Furthermore the

results show that SDM usage is linked to long-term investments. Respondents also indicated that Systems Development Methodologies are not overly complicated or hard to use and are compatible with older legacy systems; their IS department has enough experienced staff to implement Systems Development Methodologies, and so forth. Options where respondents indicated that they do not agree with (values lower than three) can be positively interpreted as well, for example: “Systems development methodologies are too complex or hard to use”. This can be interpreted that the respondent think systems development methodologies are not complex, and that they simply don’t require its implementation in their development processes.

4.3.3 Historical SDM usage

Respondents who indicated that they use SDMs in their IS department were asked to indicate how long these methodologies had been in use. The following options were available: Less than 1 year, 1-2 years, 3-5 years, 5-10 years, Over 10 years, I don’t know. The statistics for the historical usage is summarized in Figure 4.10. The results in descending order are as follows: 5-10 years (25%), I don’t know (21%), 1-2 years and Over 10 years (16%), 3-5 years (13%), and Less than 1 year (9%). As can be seen by these figures 41% of respondents have been using SDMs for over 5 years, this clearly indicates that using SDMs provided enough benefits in the long run to justify its continued use.

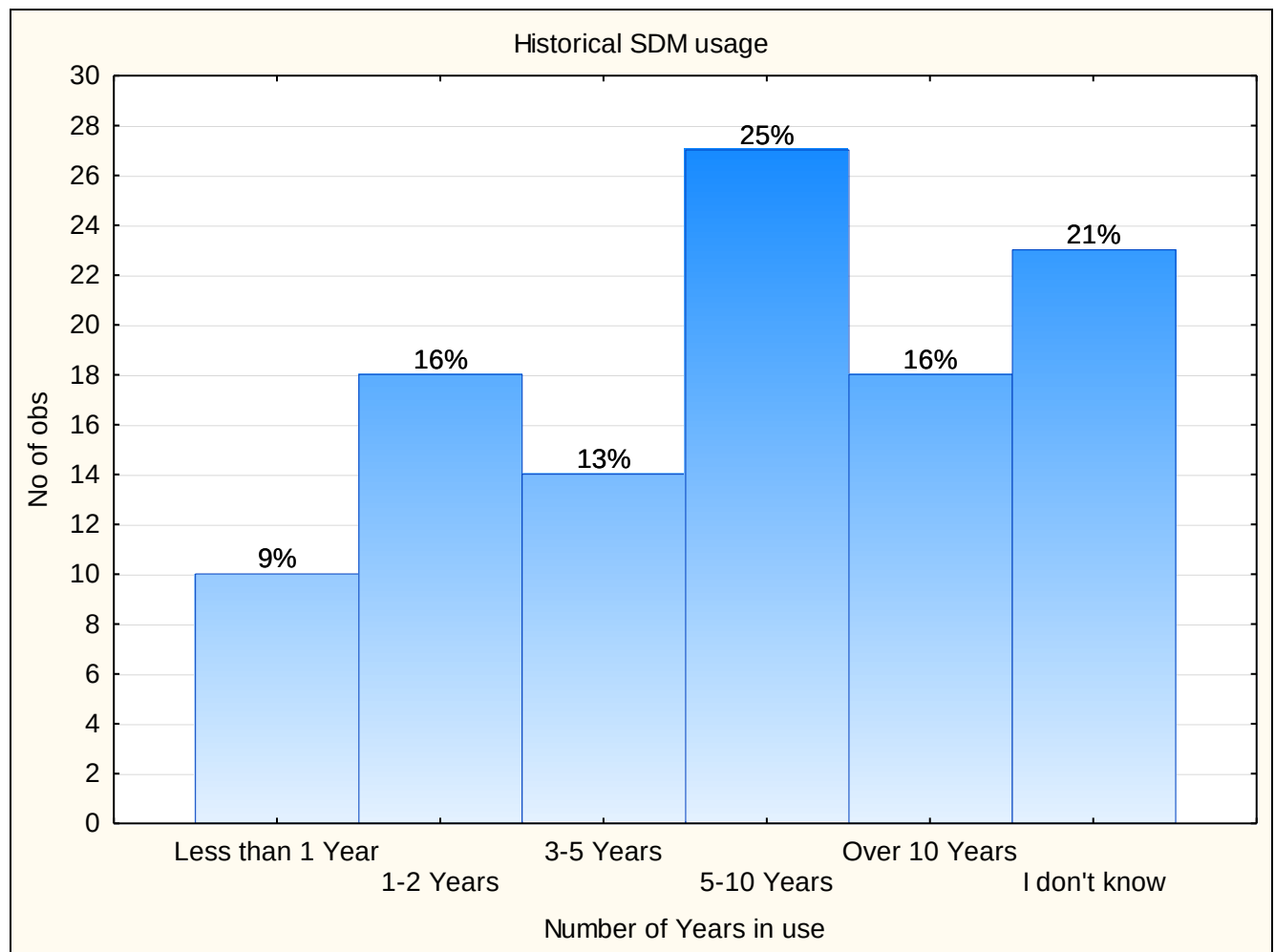


Figure 4.10 – Historical SDM usage

4.3.4 Horizontal methodology use

Horizontal use is measured across the respondent's organization. The questionnaire for this study addressed this with two questions; respondents were asked what proportions of projects (Figure 4.11) apply Systems Development Methodology knowledge. Secondly the respondents were asked what proportion of people (Figure 4.12) in their IS department apply Systems Development Methodology knowledge. The sum of these two questions was then calculated, the total was then divided by two (2) in order to get the average of the horizontal usage (Figure 4.13), the Cronbach alpha values were also computed in order to measure the reliability of the horizontal use.

-Proportion of projects:

Respondents could choose one of the following options: None, 1-25%, 26-50%, 51-75%, Over 75%. Descriptive statistics for the proportion of projects is summarized in the first part of Figure 4.10. The majority (37%) of respondents answered that over 75% of their projects were developed using SDM knowledge, this was followed by: 51-75% (27%), 26-50% (18%), 1-25% (10%), and lastly none (8%). As can be seen a total of 64% use SDMs in more than half of their projects, which indicated that they do find SDMs useful.

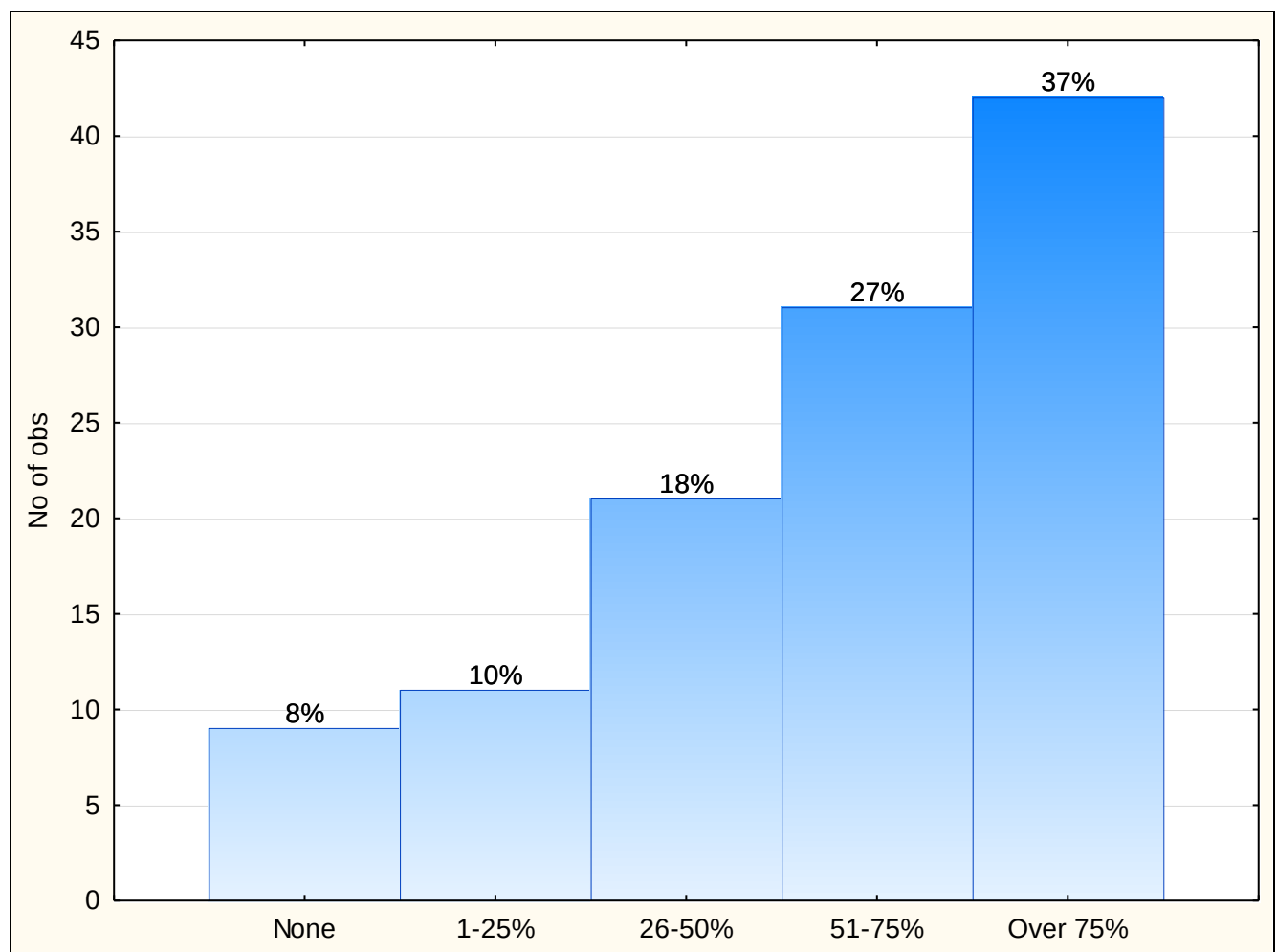


Figure 4.11 – Proportion Of Projects

Proportion of people:

As with the proportion of projects, respondents could choose one of the following options: None, 1-25%, 26-50%, 51-75%, Over 75%. Descriptive statistics for the proportion of projects is summarized in the second part of Figure 4.12. The majority (30%) of respondents answered with 51-75%, this was followed by: Over 75% (26%), 1-25% as well as 26-50% with 18% each,

lastly 7% of respondents answered with none. This is another indication that SDMs are useful seeing that in at least 56% of organisations more than half of their staff were applying SDM knowledge.

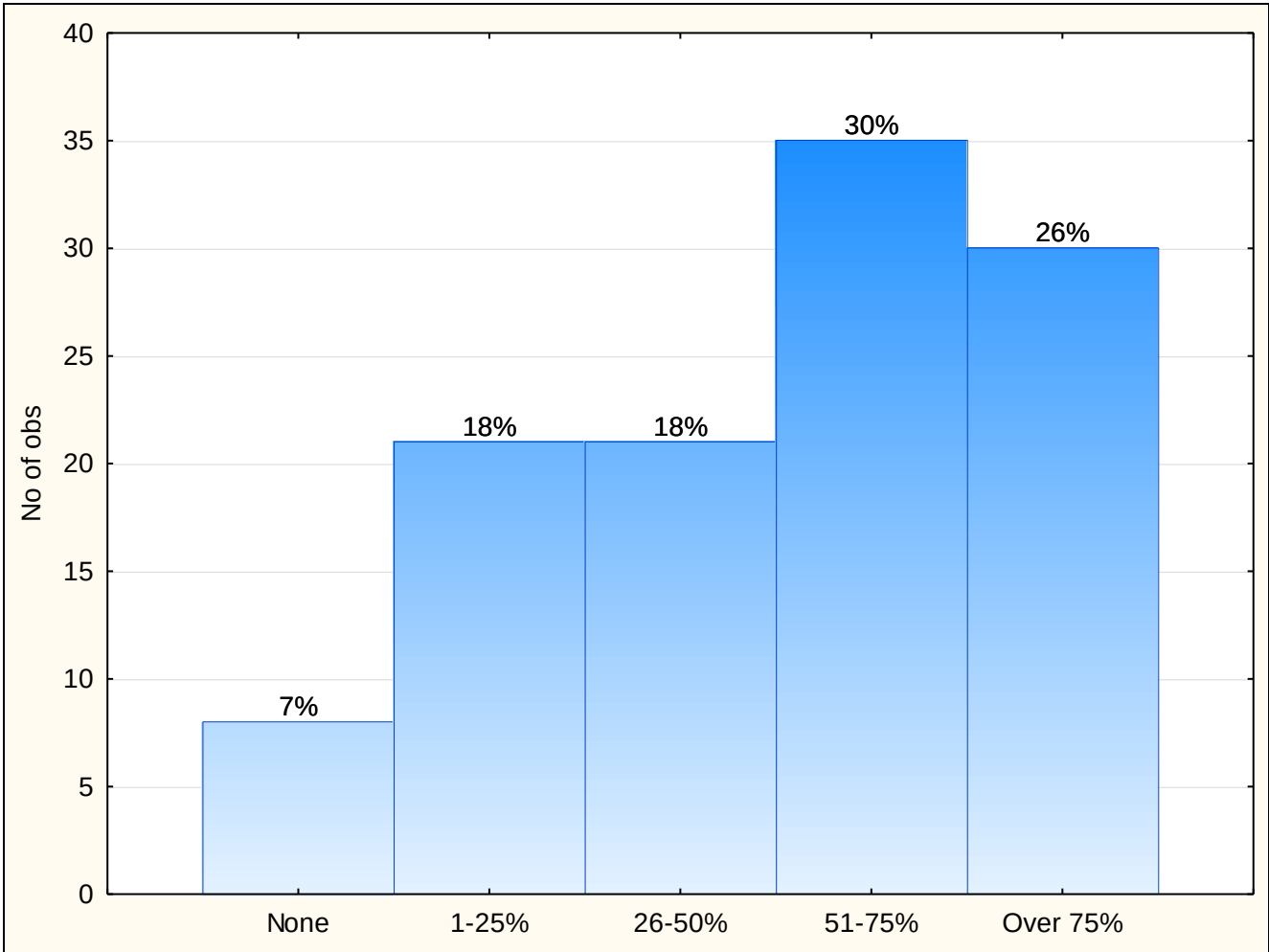


Figure 4.12 – Proportion Of People

-Horizontal Use:

As stated earlier in paragraph 4.3.2.2, the horizontal use was measured using the average of “Proportion of projects”, and “Proportion of people”. The results can be viewed in Figure 4.13. A total of 66% of the respondents indicated that they did use SDMs more than half of the time.

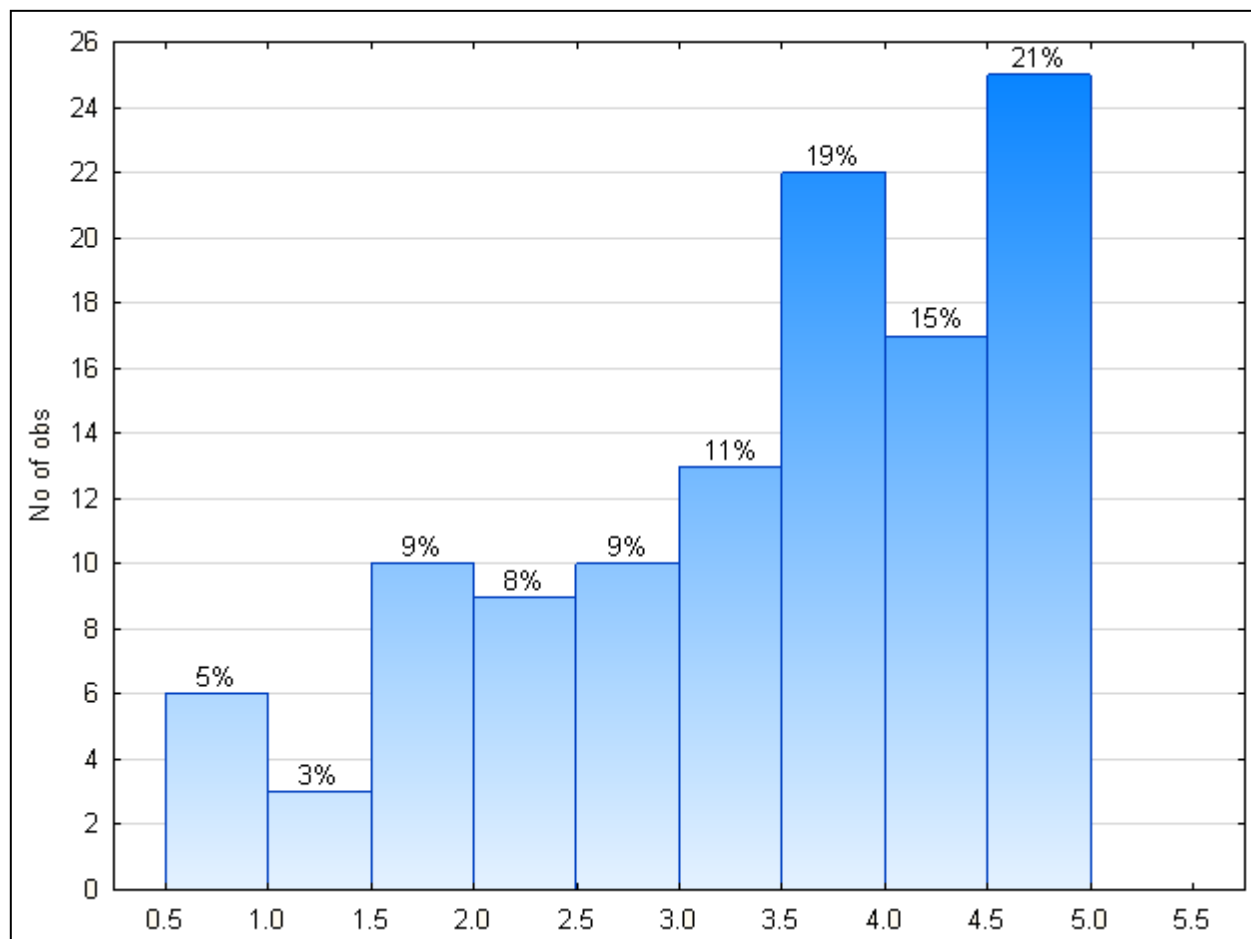


Figure 4.13 – Horizontal Use

4.3.5 Vertical methodology use

The usage intensity was measured from 1 (one) – rarely used to 5 (five) – used very often. With each respondent we used the maximum intensity of any SDM which they provided in order to determine their SDM usage intensity as a whole. The average intensity for systems development methodologies usage was 4.17, which shows that the majority of respondents did use some form of systems development methodology quite regularly to help in successfully completing a project. The list of methodologies (% of users in descending order) is shown in Table 4.2. As can be seen in Table 4.2 the single highest number of respondents are using the SDLC. If, however, the sum of the following consecutive methodologies (Extreme Programming, RAD, Agile, Information Engineering, SCRUM, ASAP, Prototyping, and RUP) are computed (which are all Agile Methodologies) it becomes clear that there is a definitive migration to newer or more Agile techniques.

Table 4.2 – Methodology usage:

	Methodology	# of Users	% of Users	Average Intensity Of Use
1	Systems Development Lifecycle	45	48.9%	3.80
2	Extreme Programming	24	26.1%	2.79
3	RAD	20	21.7%	3.70
4	Agile	18	19.6%	3.50
5	Information Engineering	13	14.1%	3.15
6	SCRUM	10	10.9%	3.90
7	STRADIS	7	7.6%	2.86
8	ASAP	6	6.5%	3.67
9	Prototyping	6	6.5%	3.33
10	RUP	5	5.4%	4.20
11	Object Oriented Systems Development	4	4.3%	4.25
12	Solution Chain Value	4	4.3%	4.00
13	Accenture Delivery Methodology	2	2.2%	5.00
14	Microsoft Agile MSF	2	2.2%	4.00
15	XPS	2	2.2%	4.00
16	SEI-PCMM 2.0	2	2.2%	3.00
17	ITIL 2.0	2	2.2%	3.00
18	SEI-CMMI for DEV 1.2	2	2.2%	3.00
19	Lean Software Development	2	2.2%	2.50
20	ABAP	1	1.1%	5.00
21	ARD	1	1.1%	5.00
22	Inhouse SDLC	1	1.1%	3.00
23	Test Driven Development	1	1.1%	4.00
24	V Model	1	1.1%	4.00
25	Vision Based Methodology	1	1.1%	3.00
26	SASDM	1	1.1%	4.00
27	DSDM	1	1.1%	3.00
28	OpenUP	1	1.1%	5.00
29	UML	1	1.1%	4.00
30	ADM	1	1.1%	3.00
31	YSM	1	1.1%	2.00
32	ASD	1	1.1%	3.00
33	PEM - Process Engineering Methodology	1	1.1%	3.00
34	RunSAP	1	1.1%	5.00
35	Crystal clear	1	1.1%	2.00

4.3.6 Strictness in regards to SDM usage

Respondents were asked to indicate how their IS departments made use of Systems Development Methodologies. Respondents could choose from one of the following options: A general guideline for all projects; a standard which is followed rigorously for all projects; and adapted on a project-to-project basis. The results indicated that 44% of respondents' IS departments adapt on a project-to-project basis. This was followed by: a general guideline for all projects with 38% and lastly a standard which is followed rigorously for all projects with 18%. The results can be viewed in Table 4.3.

Table 4.3 – SDM Usage Strictness

	% of Respondents
A general guideline for all projects	38%
Adapted on a project-to-project basis	44%
A standard which is followed rigorously for all projects	18%

As seen in Table 4.3., it is striking that merely 18% of respondents rigorously follow the SDM guidelines. Furthermore, 44% of respondents' ISD departments adapt the implementation of SDM usage on a project-to-project basis, which is referred to as contingency (applying only certain parts of systems development methodology, or only on projects which they find applicable); this is characteristic of the current era of ISD which is referred to as the "Era of methodology reassessment" by Avison and Fitzgerald (2006: 583-589).

4.4 Process Maturity Model certification

In this section we examine whether respondents are certified in CMMI, ISO 90003, or some other Maturity Model certification. Lastly we try and determine each respondent's appropriate CMMI level by analyzing their answers to the development procedures used in their IS department regardless of whether they are certified or not.

Respondents were asked whether the company they were working for/own was certified in any Process Maturity Models. The options that were available are as follows:

- CMMI,
- ISO 90003, and
- Other.

Respondents had the option of choosing either "Yes", "No", or "I don't know" for CMMI and ISO 90003. The last option ("Other") was open-ended, so respondents could write down the name of whichever Process Maturity Model they are (were) certified in. The results were as follows: for CMMI a small total of 4% of the respondents answered with "Yes", 37% answered with "No", and lastly 59% of respondents answered with "I don't know". The results for ISO 90003 were similar to those of CMMI, a total of 12% of the respondents answered with "Yes", 33% answered with "No", and lastly a total of 56% answered with "I don't know". The

statistics for CMMI can be viewed in Figure 4.14 and the statistics for ISO 90003 in Figure 4.15. None of the respondents were certified in any Process Maturity Model(s) other than the first two options that were provided.

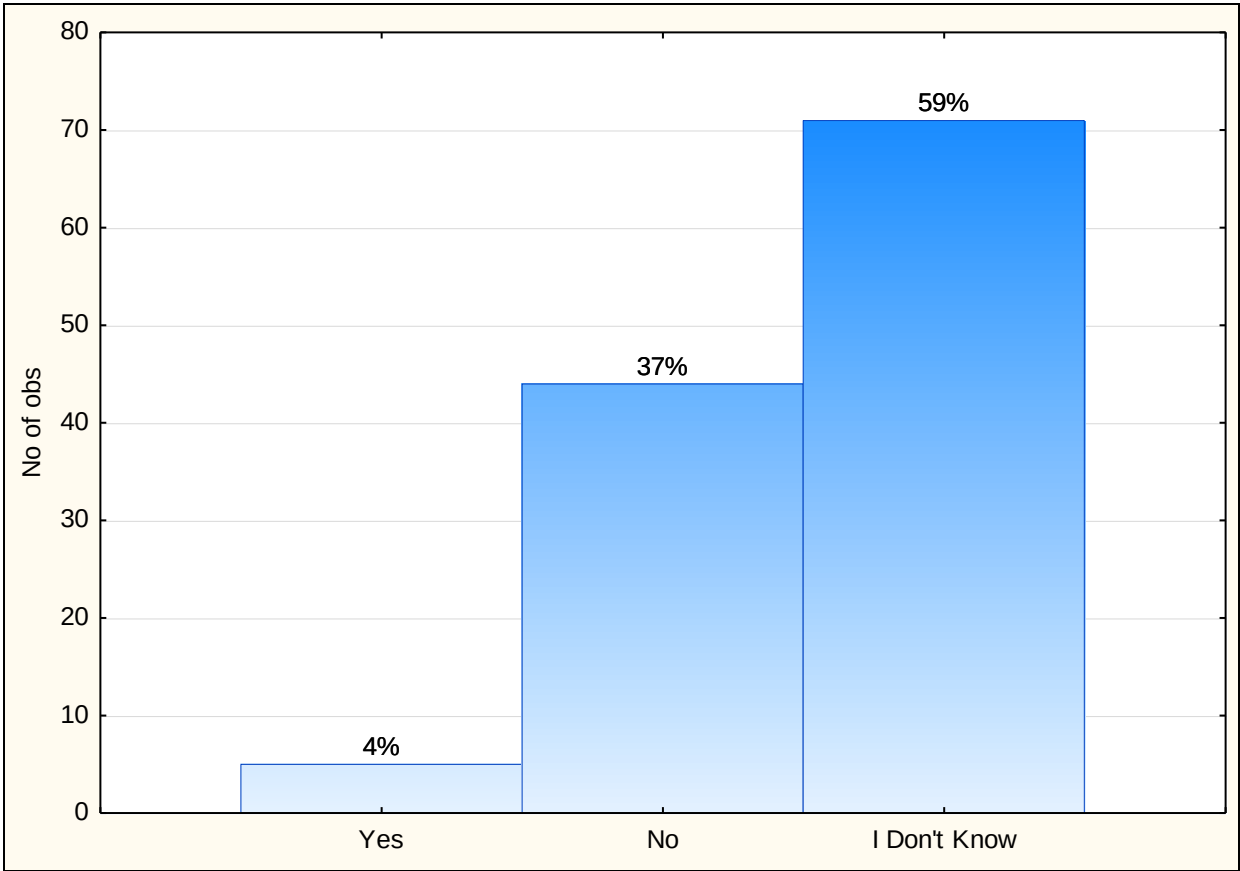


Figure 4.14 – CMMI Certification

This is an increase from 2010, where there were only two companies in South Africa which were certified in CMMI (Vera, 2010).

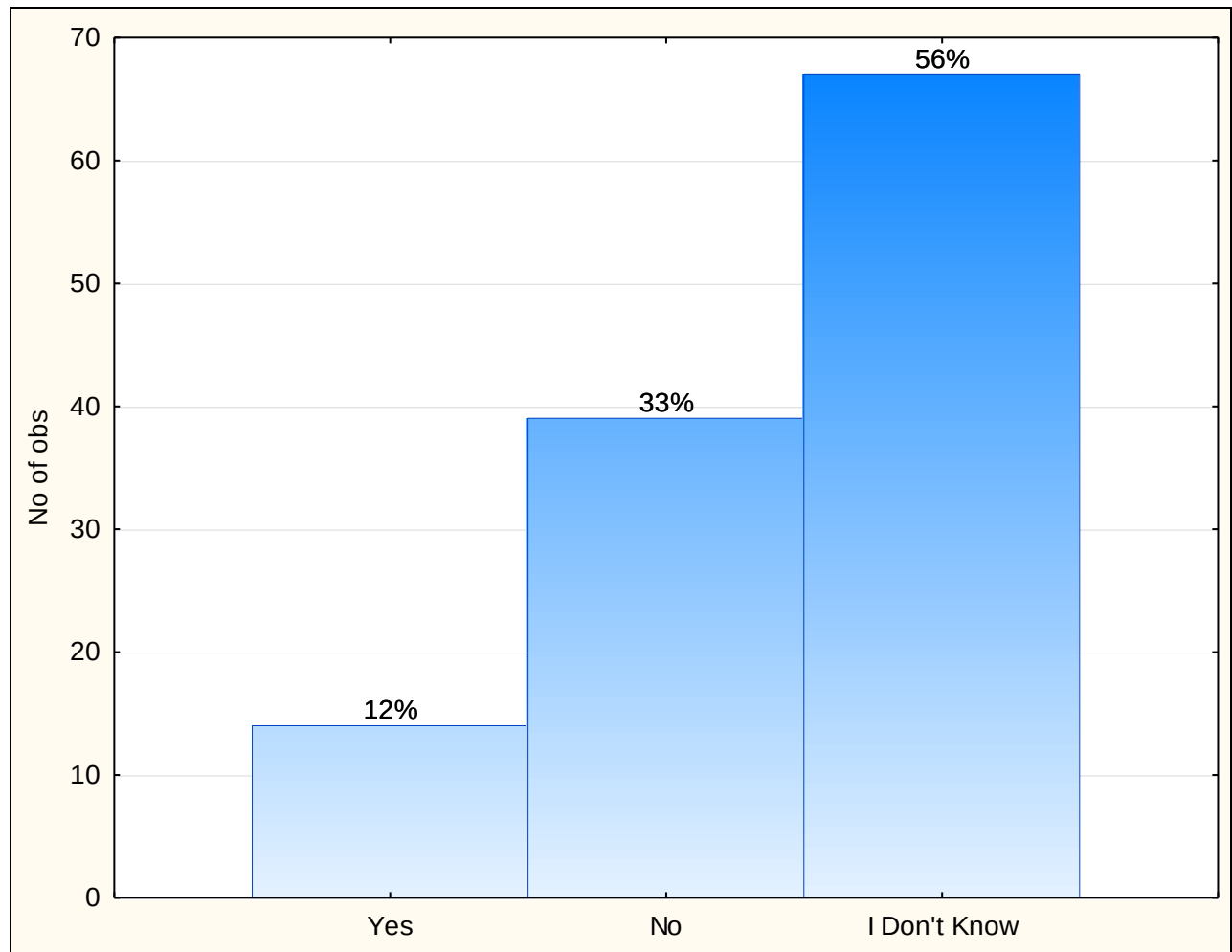


Figure 4.15 – ISO 9000 Certification

A study was done by the International Standards Organisation in 2000 indicating that 3 454 companies in South Africa were certified in ISO 9000 (ANON, 2000:12). This is significantly more than CMMI, which is also indicated by the results in Figure 4.16. This clearly indicates that the majority of respondents are not aware of whether their organisations are indeed implementing some sort of process maturity model. In order to obtain a better understanding of whether respondents are already implementing some of the requirements contained in CMMI certification an informal evaluation was also included in the research questionnaire. Within this informal evaluation each respondent's perceived CMMI level was calculated. These results are discussed below in 4.4.1.

4.4.1 Perceived CMMI Level

In order to determine the perceived CMMI level, respondents were asked to indicate which development procedures and processes are followed in their IS department. These tests were conducted due to the fact that although most organisations aren't officially certified in CMMI, they can still try and follow the processes which are required by CMMI. The full list of options is provided in Table 4.4, these options were derived from the CMMI-DEV guidelines (CMMI-DEV 1.3, 2011). Respondents could answer with one of three possible answers at each of the sub-questions. These answers were: Yes, no, and I don't know. The table below is grouped according to the appropriate CMMI level to which the question/procedure applies. Furthermore the table is ordered ascending according to the CMMI level. As discussed in the latter part of 4.4 the CMMI level displayed in Table 4.4 is the informal level assessment to which each development process/procedure applies.

Table 4.4 – CMMI Levels

Development procedures or processes	CMMI Level	% of 'YES'
Is a formal procedure used in the management review of each software development project prior to making contractual commitments?	2	79%
Is a formal procedure used to make estimates of software size?	2	69%
Is a formal procedure used to produce software development schedules?	2	74%
Is a formal procedure used to make estimates of software development cost?	2	65%
Do software development first-line managers sign off on their schedules and cost estimates?	2	69%
Does senior management have a mechanism for the regular review of the status of software development projects	2	78%
Is there a software configuration control function for each project that involves software development	2	76%
Are profiles of software size maintained for each configuration items,	2	48%

over time?		
Is a mechanism used for controlling changes to the software requirements? (Who can make changes and under which circumstances?)	2	85%
Is a mechanism used for controlling changes to the code? (Who can make changes and under which circumstances?)	2	87%
Are statistics on software code and test errors gathered?	2	50%
Does the Software Quality Assurance (SQA) function have a management reporting channel separate from the software development project management?	2	58%
Is a mechanism used for controlling changes to the software design? (Who can make changes and under which circumstances?)	3	87%
Is there a software engineering process group function?	3	53%
Does your IS department use a standardized software development process?	3	71%
Does your IS department use a standardized and documented software development process on each project?	3	64%
Is a mechanism used for ensuring compliance with the software engineering standards?	3	52%
Is there a required software engineering program for software developers?	3	49%
Is a formal training program required for design and code review leaders?	3	43%
Are internal software design reviews conducted?	3	84%
Are the action items resulting from design reviews tracked to closure?	3	69%
Are statistics on software design errors gathered?	3	43%
Are software code reviews conducted?	3	78%
Are the action items resulting from code reviews tracked to closure?	3	77%
Is there a mechanism for assuring the adequacy of regression testing?	3	58%
Is a mechanism used for verifying that the samples examined by Software Quality Assurance are truly representative of the work	3	53%

performed?		
Are design errors projected and compared to actual?	4	40%
Are the review data gathered during design reviews analysed?	4	48%
Are code and test errors projected and compared to actual?	4	40%
Are the error data from code reviews and tests analysed, to determine the likely distribution and characteristics of errors remaining in the product?	4	43%
Is design and code review coverage measured and recorded?	4	42%
Is review efficiency analysed for each project?	4	37%
Are code review standards applied?	4	62%
Is test coverage measured and recorded for each phase of functional testing?	4	66%
Has a managed and controlled process database been established for process metrics data across all projects?	4	44%
Is a mechanism used for periodically assessing the software engineering process, and implementing indicated improvements?	4	50%
Are analyses of errors conducted to determine their process related causes?	4	61%
Is a mechanism used for managing and supporting the introduction of new technologies?	4	68%

As seen in the table there were a total of:

- Twelve (12) x Level 2 processes.
- Fourteen (14) x Level 3 processes.
- Ten (10) x Level 4 processes.

From the data analysis the following became apparent: although the majority of companies aren't certified in CMMI, the respondents felt that they do in fact implement some of the key-practices required to be certified in CMMI. The average number of the activities performed for all respondents are summarised in Table 4.5, as well as the perceived CMMI Level. Firstly Table 4.5 shows the perceived CMMI level; secondly the average number of activities for all respondents was calculated by using the following formula:

T - Number of activities per respondent for the perceived level

N - Number of respondents which completed this part of the questionnaire

This formula can be summarised as follows:

$$\left(\sum T\right) \div N$$

The third column shows the maximum number of activities for the specific level and lastly the fourth column summarises the average total number of activities completed by dividing the “Number of Activities” through the “Max Number of Activities”.

Table 4.5 – Perceived CMMI Level

CMMI Level	Number of Activities	Max Number of Activities	% of Activities
2	7.76	12	65%
3	8.78	14	63%
4	5.98	10	50%

Furthermore a Tree Clustered Analysis was done on the different activities to help in grouping the number of activities completed into a perceived CMMI Level. As seen from the results in Figure 4.16, most respondents can be grouped into a Level 2/3 group. When comparing the Euclidian Distances, Levels 2 and 3 are linked at a distance of 196, these two groups are then joined by Level 4 at a distance of 292.

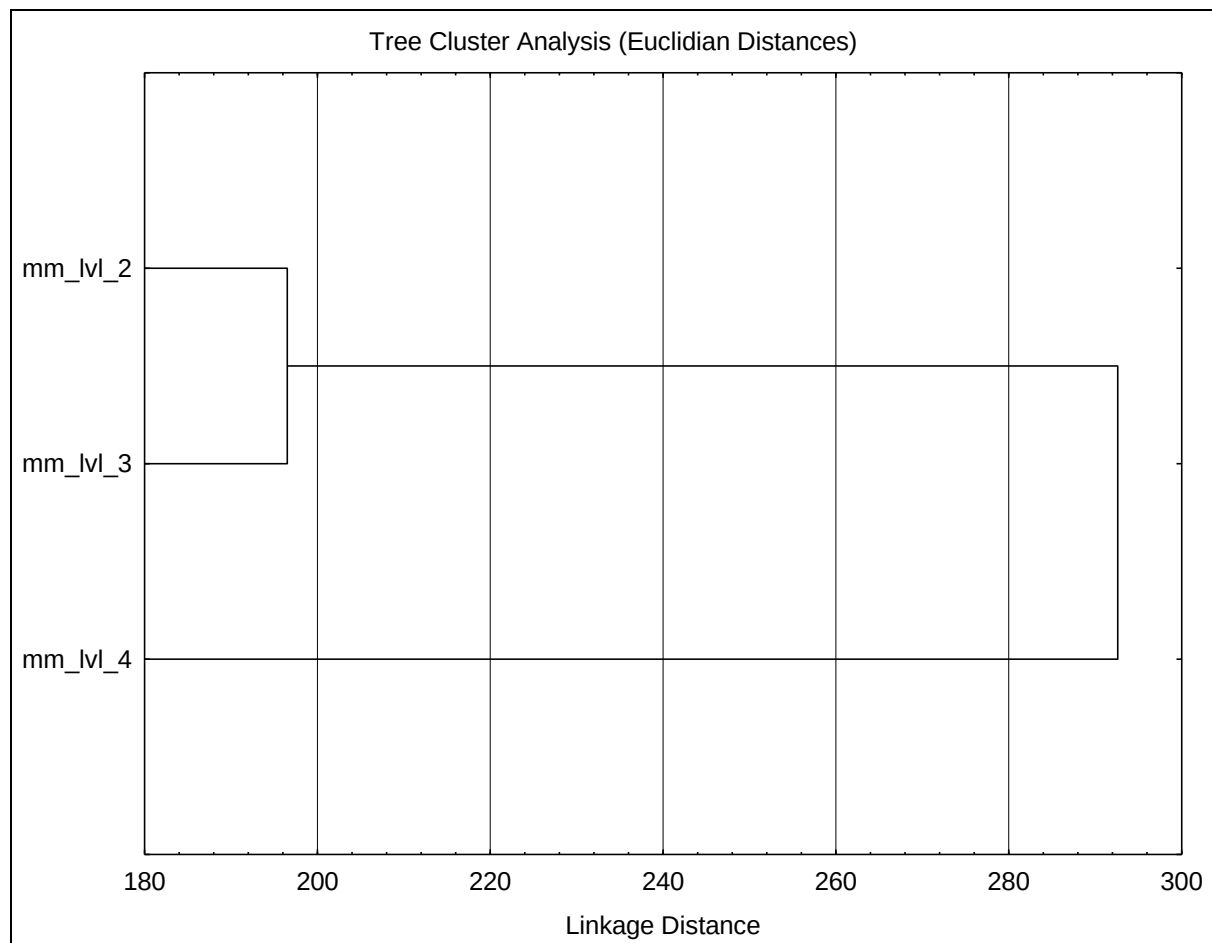


Figure 4.16 – Tree Cluster Analysis on CMMI activities.

As these results indicate, the average respondent does in fact adhere to 50% or more of the required CMMI procedures/processes which we tested for. The majority of respondents can be grouped into a Level 2/3 group, whereas some respondents can be grouped into a CMMI Level 4 group. The questions that arise from this then are whether implementation of SDMs encourages the usage of techniques required by CMMI and then lastly but most importantly how this affects project success.

4.5 Relationships between PMM certification and SDM usage success factors

In this research factors were identified which could have certain relational influence on each other. In this section we will examine these relationships in an attempt to be able to predict factors such as project or process success. Multiple regression was used to determine whether the relationships were significant or not. In order to view a relationship's significance the p-value was examined. The p-value scale that was used is as follows:

- $p < 0.1'$ - Noteworthy
- $p < 0.05^*$ - Significant
- $p < 0.01^{**}$ - Very significant
- $p < 0.001^{***}$ - Extremely significant

The values reported are the beta (β) values which accompany the p-values.

4.5.1 Information Systems Development Success

In the following discussions the success of each respondent's Information Systems Development was examined. The research focused on the two main parts of Information Systems Development namely; the success of the development process as well as the success of the developed product.

4.5.1.1 Success of the IS Development Process

In order to measure the success of the IS development process, the respondents were asked to indicate to what extent they agreed with the statements in the first column of Table 4.6. Possible answers varied from 1 (one) to 5 (five), whereas 1 (one) was totally disagree and 5(five) totally agree. The results can be reviewed in Table 4.6.

Table 4.6 – IS Development process success

#	Question	Average
7	The project achieved its goals.	4.2
9	The project was a success.	4.1
8	The project represents excellent work.	4
3	The developed systems satisfied all the stated requirements.	3.9
5	The productivity of the developers involved with the project was high.	3.9
2	The project was completed within the budget.	3.7
4	The speed of developing the information system was high.	3.6
1	The project was completed on schedule.	3.5
6	The cost of the project is low when compared to the size and complexity of the system developed.	3.3

As can be seen by reviewing the top-rated answers; respondents felt that their last Information Systems project had been a success and that the system met its requirements due to their excellent work.

Success of the development process was measured by using the instrument suggested by Huisman (2000). It included 9 items. The reliability of this 9-item measurement was 0.86. Factor analysis regarding the success of the development process indicated that all of the items loaded on one factor, with the explained variance = 4.58 and the proportional total = 0.51. Therefore, the average for the 9 items was used to measure the success of the development process.

Factor analysis using the principal-components method with varimax normalised rotation was performed on the data. The rudimentary idea behind factor analysis is to express two or more variables through a single factor. In order to determine the number of factors to retain, the Kaiser criterion was used. This criterion is widely used and states that only factors with Eigen values (variances extracted by each factor) greater than 1 are retained.

The reliability of these factors is very important. Each factor should be a reliable measure of its corresponding research variable. Reliability is the extent to which an instrument is free of measurement errors (Conger, 1994). A measurement is reliable if it mostly reflects a true score, relative to the error. This means that if a reliable instrument were given to the same group of people on multiple occasions, the same answers would be obtained. Reliability analysis was performed on the items of each of the factors identified using Cronbach's coefficient alpha as the index of reliability, because it is used most commonly. If all items in the factor are perfectly reliable and measure the same thing, then Cronbach's coefficient alpha is equal to 1. There is no exact threshold for reliability, but researchers have resorted to using a Cronbach alpha of 0,6 to estimate reliability (Roberts et al., 1998; Huisman, 2000). For exploratory research studies, it is agreed that a coefficient alpha level of 0.5 could be deemed acceptable (Nunally, 1967).

To measure the reliability of the ISD process success; a Cronbach Alpha test was conducted on the data. The results can be reviewed in Table 4.7

Table 4.7 – Reliability Analysis for IS Development process success

Construct	Question Numbers (Appendix A)	Reliability
Process Success	Question 11 Numbers 1-9	0.865

As can be seen the reliability was high which confirms findings mentioned above.

4.5.1.2 *Success of the IS Developed Product*

In order to measure the success of the developed product, the respondents were asked to indicate to what extent they agreed with the statements in regards to the success of the developed product. In the first column of Table 4.7 the relevant question number is indicated. The second column of Table 4.7 indicates the possible answers which varied from 1 (one) to 5 (five), whereas 1 (one) was totally disagree and 5(five) totally agree. The results can be reviewed in Table 4.8.

Table 4.8 – IS developed product success

#	Question	Average
1	The functionality of the developed system is high	4.1
11	The developed system is a success.	4.1
2	The reliability of the developed system is high.	4
6	The usability of the developed system is high.	4
7	The developed system meets the user needs.	4
9	The quality of the developed system is high.	4
5	The efficiency of the developed system is high.	3.9
10	The users are satisfied with the developed system.	3.9
3	The maintainability of the developed system is high.	3.8
4	The portability of the developed system is high.	3.2
8	The documentation of the developed system is good.	3.1

By reviewing the abovementioned data, one can see that respondents focused on delivering a product that complied with its initial expectations and that functionality was rated higher than a well-documented product.

Success of the developed system was measured by using the instrument suggested by Huisman (2000). It included 11 items. The reliability of this 11-item measurement was 0.90. Factor analysis regarding the success of the developed system indicated that all of the items loaded on one factor, with the explained variance = 5.86 and the proportional total = 0.53. Therefore, the average for the 11 items was used to measure the success of the development process.

To measure the reliability of the ISD product success; a Cronbach Alpha test was conducted on the data. The results can be reviewed in Table 4.9

Table 4.9 – Reliability Analysis for IS Development product success

Construct	Question Numbers (Appendix A)	Reliability
Product Success	Question 12 Numbers 1-11	0.898

The abovementioned confirms that the data used is reliable and confirms findings.

Horizontal methodology use refers to the manner in which the methodology is used across the organisation. It was measured using two items, namely the proportion of projects that are developed in the IS department by applying systems development methodology knowledge, and the proportion of people in the IS department that apply systems development methodology knowledge regularly. These two items reveal whether the systems development methodology knowledge was applied widely in the organisation. The reliability of these two items was 0.87.

Lastly a Cronbach Alpha test was also conducted on the horizontal use. The results can be reviewed in Table 4.10.

Table 4.10 – Reliability Analysis

Construct	Question Numbers (Appendix A)	Reliability
Horizontal Use	Questions 16 and 17	0.870

The above-mentioned table indicates that SDM knowledge was applied and contributed towards a higher ISD process success as well as a higher ISD product success.

4.5.2 Factors which influence the relationship between the IS development process- and the IS developed product success

The relationship between SDM usage (horizontal use, vertical use, usage strictness, and historical use) and the success of the development process, as well as the relationship between the SDM usage (horizontal use, vertical use, usage strictness, and historical use) and the developed product success, was measured in order to determine whether using an SDM will actually help increase the quality of the development process and the developed product. Furthermore the relationship between the perceived CMMI levels and the success of the process as well as the success of the product was measured. The latter helped in determining whether being certified in CMMI increases the success rates of either the development process, or the developed product.

Firstly the results indicated that an extreme significance exists between the development process success and Horizontal SDM Usage, as well as the developed product success and Horizontal SDM Usage. This indicates that if more employees, or more projects, were to implement SDM knowledge, both the process- and product success would increase. The statistics reviewing the results is summarized in Table 4.11. In the first column the usage factors are listed; the second column contains the relationships between the process success and the usage factors, and lastly the third column contains the relationships between the product success and the usage factors. As stated in paragraph 4.11, the values of the usage factors are displayed using the appropriate β (beta) values.

Table 4.11 –Results of the multiple regression analysis depicting the relationship of SDM usage and process- as well as product success.

	Process Success	Product Success
SDM Usage Intensity (Vertical)	0.02	0.03
SDM Usage (Horizontal)	0.38***	0.42***
SDM Time in use	0.11	0.16
SDM Usage Strictness	0.02	-0.08
R	0.43	0.51
R ²	0.19	0.26
Adjusted R ²	0.15	0.22
F	4.95	7.47

' $p \leq 0.10$ * $p \leq 0.05$ ** $p \leq 0.01$ *** $p \leq 0.001$

Secondly, results indicate that a significant relationship exists between CMMI Level 4 processes/procedures and the development process success. This shows that the processes/procedures required in attaining CMMI Level 4 certification help in improving the quality/success of the development process, the appropriate regression statistics can be viewed in Table 4.12. The first column contains the different perceived CMMI levels, the second column contains the relationships between process success and the perceived CMMI levels, and lastly the third column contains the relationships between product success and the perceived CMMI levels. As stated in paragraph 4.11, the values of the perceived CMMI levels are displayed using the appropriate β (beta) values.

Table 4.12 – Results of multiple regression analysis depicting the relationship of the perceived CMMI Levels and process- as well as product success.

	Process Success	Product Success
CMMI Level 2	-0.05	0.06
CMMI Level 3	-0.09	-0.21
CMMI Level 4	0.27*	0.24
R	0.20	0.17
R ²	0.04	0.03
Adjusted R ²	0.13	0.004
F	1.52	1.16

' $p \leq 0.10$ * $p \leq 0.05$ ** $p \leq 0.01$ *** $p \leq 0.001$

4.6 Relationships of the Horizontal- and Vertical SDM Usage, SDM Usage Strictness, and the SDMs Time in Use in regards to the perceived CMMI Levels

These relationships were examined in order to determine whether using SDMs results in a higher perceived CMMI level. As seen in Table 4.13 the results indicate that if an SDM has been in use for a longer period of time, then it is more likely that a higher CMMI level can be obtained. In Table 4.13 the first column contains the perceived CMMI levels, the second column contains the relationships between the perceived CMMI levels and the Horizontal SDM Usage, the third column contains the relationships between the perceived CMMI levels and the Vertical SDM Usage, the fourth column contains the relationships between the perceived CMMI levels and the SDM Usage Strictness, and lastly the fifth column contains the relationships between the perceived CMMI levels and the SDM Historical use. As stated in paragraph 4.11, the values of the perceived CMMI levels are displayed using the appropriate β (beta) values.

Table 4.13 - Results of multiple regression analysis depicting the relevance of SDM Usage factors and the perceived CMMI levels.

	SDM Usage (Horizontal)	SDM Usage (Vertical)	SDM Usage Strictness	SDM Time in use
CMMI Level 2	0.08	0.02	0.10	0.09
CMMI Level 3	-0.24	0.09	-0.23	-0.12
CMMI Level 4	0.22	-0.12	0.08	0.40**
R	0.16	0.09	0.13	0.37
R ²	0.25	0.01	0.02	0.14
Adjusted R ²	-	-	-	0.11
F	0.75	0.21	0.48	4.54

4.7 Chapter Summary

Chapter 4 focused on the research findings; the data was reviewed and analysed statistically. Each respondent's development environment was analysed, ranging from background information such as the size of their organisation to the outcome their last IS project. Furthermore SDM usage and PMM certification was examined and to summarise we reviewed how various factors influenced the success of the development process as well as the success of the developed product.

Chapter 5 – Summary and Final Research Conclusions

5.1 Introduction

This chapter focuses on the research results, conclusions, and contributions of this study. Firstly we review our research aims and objectives, and then we examine the results of this study and summarize its contributions. A brief comparative study of previous research has been done in regards to this study. We then state any limitations that were experienced with this research or limitations to its implications. This is followed by suggesting a few ideas which can be considered for future research, and then lastly we summarize and review what was said in this chapter.

5.2 Research Results and contributions

The key research aims and objectives were listed in Chapter 1. To review, these aims were:

1. Attain background information on organizations' usage of Systems Development Methodologies during the development of information systems.
2. Determine each organization's perceived Process Maturity level. Grouped organisations in to their approximate CMMI level by conducting informal tests.
3. Determine each organisation's Process Maturity Model usage.
4. Determine the relationship (if any) between the Process Maturity level of the organisation and the use of Systems Development Methodologies.
5. Determine the relationship (if any) between the Process Maturity level of the organization and the effectiveness of Systems Development Methodologies.
6. Determine the relationship between the organisation's development process success and its use of Systems Development Methodologies.
7. Determine the relationship between the organisations' developed product success and its use of Systems Development Methodologies.

In Chapter 4 the research results indicated the following concerning the research aims and objectives:

1. Systems Development Methodology usage

- A total of 73% of the respondents indicated that they do use Systems Development Methods to some extent. The top reasons why respondents felt that the implementation of Systems Development Methodologies was not required include: Their current development process is adequate and they required short term benefits whereas they felt that SDMs only provide long term benefits.
- Out of those indicating that they do use Systems Development Methodologies the majority (37%) implement it in more than 75% of their projects. To continue 56% of respondents also indicated that 51% or more of employees involved in the development process do use some form of Systems Development Methodology knowledge on a regular basis.
- Lastly respondents indicated that the majority (44%) adapt Systems Development Methodology usage and implementation on a project-to-project basis.
- The top used methodologies along with its average use intensity include; Systems Development Lifecycle (3.8), Extreme Programming (2.79), RAD (3.7), Agile (3.5), and Information Engineering (3.15). For further information refer to Tale 4.2.

2. Perceived Process Maturity Level

- The majority (59%) of respondents, when asked, indicated that they were unsure whether their organization was certified in the Capability Maturity Model Integration, and 56% were also unsure whether their organization was certified in ISO 90003.
- Out of the 125 respondents, the perceived CMMI levels were quite average for the majority. The highest perceived level (percentage wise) was Level 2 with an average of 65% of activities being satisfied.

3. Relationship between perceived CMMI level and Systems Development Methodology usage

- No significant statistical evidence was found in regards to the perceived CMMI level and the horizontal, vertical and strictness of use of Systems Development Methodologies.

- There was however a very significant indication of a larger likelihood for satisfying more of the CMMI's Level 4 activities in organizations which have been using Systems Development Methodologies for a longer period of time.
4. The effect of implementing Systems Development Methodologies and Process Maturity Models
- Results indicate that the Horizontal use (number of projects/people which implement SDM knowledge) of Systems Development Methodologies have an extremely significant effect on the development process- and the developed product success.
 - The results also indicate that respondents who are satisfying more of CMMI's Level 4 activities have a higher development process success, which results in a higher quality development process.

Research results also indicated number of other significant factors; some of these can be summarized as follow:

- Most respondents' organizations are not certified in some formal Process Maturity Model, even though they do in fact use/implement the Software Process Improvement (SPI) techniques to some extent.
- The majority of respondents which do implement some form of Systems Development Methodology knowledge have been doing so for three years or more.
- The most common Systems Development Methodology type, when using Avison and Fitzgerald's framework for classifying Methodologies (Avison & Fitzgerald, 2006:568), is Agile Methodologies. This is an indication of more companies progressing from the traditional Systems Development Lifecycle (SDLC) to newer/quicker development methods which is better suited to handle changes in scope or requirements (which is a typical characteristic of Agile Methodologies).

5.3 Limitations

Factors which limited this research's true potential included the unresponsiveness of people contacted/approached to complete the accompanying questionnaire. The low number of CIOs (Chief Information Officer's) or other managerial input could be one of the reasons why such a low number of respondents indicated being certified in some form of Process Maturity Model.

Both CMMI and ISO 90003 have a vast collection of requirements which can't be fully summarized into a short questionnaire form without compromising some content.

5.4 Conclusions

Some of the conclusions which can be made from this dissertation can be summarized as follows: Most companies in South Africa, whether specializing in Software Development or any other area, have no official Process Maturity certification. Most of these companies however, does in fact use some form of a standardised development method(s). These methods (often some implementation of a formal Systems Development Methodology) help in increasing the quality as well as the success of the developed product and the development process. Techniques used (required) in these Systems Development Methodologies also aid companies in becoming certified by implementing/requiring some of the standards which Process Maturity Models such as CMMI require. All of these factors play a role in increasing the quality of the organisation's success; which in the end leads to a larger profit margin, increased user satisfaction, increased employee satisfaction, and so forth.

5.5 Future research

Further suggestions for future research can be as follows:

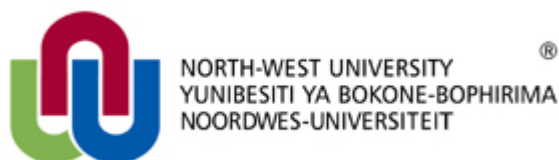
- Focus on obtaining at least ten (10) individual responses per organization to form a holistic view of each responding organization.
- Research other newer Process Maturity Models.
- Attempt to determine whether a relationship exists between certain types of methodologies and Process Maturity Model certification.
- Expand to include respondents from other countries.

5.6 Closing

As this chapter and especially this research come to a close, a brief summary should be made. This chapter briefly discussed previous research, research results and contributions, limitations, and lastly possible research which can be done in the future. The goals of this research were to determine whether a relationship exists between Process Maturity Models and the use and effectiveness of Systems Development Methodologies. A question which is

examined in further detail, analysed, and then answered throughout this study is that a relationship exists between Process Maturity Models, the use and effectiveness of Systems Development Methodologies and the quality/success of the development process as well as the developed product.

Appendix A – Research Questionnaire



North-West University
Potchefstroom Campus

Private bag X6001 Potchefstroom 2520
Tel (018) 299 1111 Fax (018) 299 2799
<http://www.nwu.ac.za>

Computer Science and Information Systems

Tel (018) 299-2537
Fax (018) 299-2570
E-Mail 20385463@nwu.ac.za

August 2010

1. In which country do you live? - Please mark the applicable answers with an X	
Australia	
Finland	
Ireland	
Malaysia	
South Africa	
United States of America	
Other, please specify	

2. What is the core business area of your organization?	
Agriculture	1
Catering, accommodation and other trade	2
Construction	3
Education	4
Finance/Banking/Insurance	5
Manufacturing	6
Retail, motor repair services	7
Software house/Software consulting	8
Wholesale trade & commercial agents	9
Transport, storage and communication	10
Other, please specify	11

3. How long has this business been in operation?	
Less than 1 year	1
1-3 years	2
3-6 years	3
6-9 years	4
More than 9 years	5

4. What is the general skill of your employees involved in information system development (ISD)?	
No information system development skills	1
Limited skills	2
Fairly skilled	3
Well skilled	4
Experts	5

5. What is the total number of people employed in your business (total from all locations)?	
1-5	1
6-20	2
21-50	3
51-100	4
101-200 (or more)	5

6. What is the total number of people who work in this business' system development department?	
0 (none)	1
1-5	2
6-10	3
11-20	4
21-50	5
More than 50	6

7. What is your role within this business?	
Owner	1
Executive Director	2
Manager	3
Project Leader	4
Developer	5
Other, please specify:	6

8. How does your business procure software? (You may mark more than one)	
We do not procure any software	1
In-house development	2
Off-shelf (no customization)	3
Off-shelf (with customization)	4
Out-sourcing	5
Other, please specify:	6

9. Which one of the following describes the outcome of the last information system development project you were involved with	
I have not partaken in any information system development projects.	1
The project was cancelled/terminated before completion	2
The project was completed but not implemented	3
The project was completed and implemented but not in use any more	4
The project was completed, implemented and is still in use	5

10. What is the size of the last information system development project you were involved with?	
Small	1
Medium	2
Large	3
Very large	4
I have not partaken in any information system development projects	5

11. To what extent do you agree/disagree with the following statements about the last information system development project you were involved with? (1 = totally disagree, 5 = totally agree)

	Totally disagree			Totally agree	
11.1) The project was completed on schedule	1	2	3	4	5
11.2) The project was completed within the budget	1	2	3	4	5
11.3) The developed system satisfied all the stated requirements	1	2	3	4	5
11.4) The speed of developing the information system was high	1	2	3	4	5
11.5) The productivity of the developers involved with the project was high	1	2	3	4	5
11.6) The cost of the project is low when compared to the size and complexity of the system developed.	1	2	3	4	5
11.7) The project achieved its goals	1	2	3	4	5
11.8) The project represents excellent work	1	2	3	4	5
11.9) The project was a success	1	2	3	4	5

12. To what extent do you agree/disagree with the following statements about the last information system development project you were involved with? (1 = totally disagree, 5 = totally agree)

	Totally disagree			Totally agree	
12.1) The functionality of the developed system is high	1	2	3	4	5
12.2) The reliability of the developed system is high	1	2	3	4	5
12.3) The maintainability of the developed system is high	1	2	3	4	5
12.4) The portability of the of the developed system is high	1	2	3	4	5

12.5) The efficiency of the developed system is high	1	2	3	4	5
12.6) The usability of the developed system is high	1	2	3	4	5
12.7) The developed system meets user needs	1	2	3	4	5
12.8) The documentation of the developed system is good	1	2	3	4	5
12.9) The quality of the developed system is high	1	2	3	4	5
12.10) The users are satisfied with the developed system	1	2	3	4	5
12.11) The developed system is a success	1	2	3	4	5

13. If you are NOT using systems development methodologies please indicate to what extent do you agree/disagree with the following statements about the last information system development project you were involved with? (1 = totally disagree, 5 = totally agree)

	Totally disagree			Totally agree	
13.1) The profile of development projects in our IS department doesn't require the use of system development methodologies	1	2	3	4	5
13.2) System development methodologies are too complex or hard to use	1	2	3	4	5
13.3) The current system development practice in our IS department is adequate	1	2	3	4	5
13.4) The experience of the developers in our IS department reduces the need for systems development methodologies	1	2	3	4	5
13.5) The benefits of systems development methodologies use are long-term, whereas cost are incurred short term	1	2	3	4	5
13.6) There is a lack of experienced staff in our IS department who can effectively use system development methodologies	1	2	3	4	5
13.7) New systems developed with systems development methodologies are not compatible with legacy systems	1	2	3	4	5
13.8) Our IS department lacks a suitable environment to support systems development methodologies	1	2	3	4	5
13.9) In our IS department there is a lack of management support for the use of systems development methodologies	1	2	3	4	5
13.10) The learning curve for systems development methodologies are very long	1	2	3	4	5
13.11) The financial investment in systems development methodologies is to large	1	2	3	4	5
13.12) In our IS department there is a lot of uncertainty over the benefits of adopting systems development methodologies.	1	2	3	4	5
13.13) In our IS department there is no clear	1	2	3	4	5

objectives for adopting systems development methodologies					
---	--	--	--	--	--

14. There are various types of system development methodologies, for example XP (Extreme programming) and IE (Information Engineering). Name the methodologies your organization uses, and indicate how intensively each one is used: (1 = used very infrequently, 5 = used very often)

Name of Methodology	Intensity of use				
	1	2	3	4	5
	1	2	3	4	5
	1	2	3	4	5
	1	2	3	4	5
	1	2	3	4	5
	1	2	3	4	5
	1	2	3	4	5
	1	2	3	4	5
	1	2	3	4	5

15. How long has your systems development methodology been in use in your IS department?	
Less than 1 year	1
1-2 years	2
3-5 years	3
5-10 years	4
Over 10 years	5
I don't know	6

16. What is the proportion of projects that are developed in your IS department by applying systems development methodology knowledge?	
None	1
1 – 25 %	2
26 – 50 %	3
51 – 75 %	4
Over 75 %	5

17. What is the proportion of people in your IS department that apply systems development methodology knowledge regularly?	
None	1
1 – 25 %	2
26 – 50 %	3
51 – 75 %	4
Over 75 %	5

18. Which of the following best describes how your IS department makes use of its systems development methodologies?	
A general guideline for all projects.	1
A standard which is followed rigorously for all projects.	2
Adapted on a project –to-project basis.	3

19. Is your business currently certified in:			
CMMI	Yes	No	I don't know
ISO 9000-3	Yes	No	I don't know
Other (please specify): - - -			

If you are answered “Yes” to any of question number 19’s options, please complete the following question:

20. Why did you want to become certified? - please mark the applicable answers with an X	
Better market impression	
Better quality products	
Faster development times	
The required skills/tools are already available	
Other (please specify): - - -	

If you answered “No” to all of question number 21’s options, please complete the following question:

21. Why are you not currently certified? - please mark the applicable answers with an X	
Did not meet required criteria	
We see no possible advantages	
Insufficient funding	
The required skills/tools are not available	
Other (please specify): - - -	

22. Please describe the development procedures and processes in your IS department.		
22.1) Is a formal procedure used in the management review of each software development project prior to making contractual commitments?	YES	NO
22.2) Is a formal procedure used to make estimates of software size?	YES	NO
22.3) Is a formal procedure used to produce software development schedules?	YES	NO
22.4) Is a formal procedure used to make estimates of software development cost?	YES	NO
22.5) Do software development first-line managers sign off on their schedules and cost estimates?	YES	NO
22.6) Does senior management have a mechanism for the regular review of the status of software development projects	YES	NO
22.7) Is there a software configuration control function for each project that involves software development	YES	NO
22.8) Are profiles of software size maintained for each configuration items, over time?	YES	NO
22.9) Is a mechanism used for controlling changes to the software requirements? (Who can make changes and under which circumstances?)	YES	NO
22.10) Is a mechanism used for controlling changes to the software design? (Who can make changes and under which circumstances?)	YES	NO
22.11) Is a mechanism used for controlling changes to the code? (Who can make changes and under which circumstances?)	YES	NO
22.12) Is there a software engineering process group function?	YES	NO
22.13) Does your IS department use a standardized software development process?	YES	NO
22.14) Does your IS department use a standardized and documented software development process on each project?	YES	NO
22.15) Is a mechanism used for ensuring compliance with the software engineering standards?	YES	NO
22.16) Is there a required software engineering program for software developers?	YES	NO
22.17) Is a formal training program required for design and code review leaders?	YES	NO
22.18) Does the Software Quality Assurance (SQA) function have a management reporting channel separate from the software development project management?	YES	NO
22.19) Are internal software design reviews conducted?	YES	NO
22.20) Are the action items resulting from design reviews tracked to closure?	YES	NO
22.21) Are statistics on software design errors gathered?	YES	NO
22.22) Are software code reviews conducted?	YES	NO
22.23) Are the action items resulting from code reviews tracked to closure?	YES	NO
22.24) Are statistics on software code and test errors gathered?	YES	NO
22.25) Are design errors projected and compared to actual?	YES	NO
22.26) Are the review data gathered during design reviews	YES	NO

analysed?		
22.27) Are code and test errors projected and compared to actual?	YES	NO
22.28) Are the error data from code reviews and tests analysed, to determine the likely distribution and characteristics of errors remaining in the product?	YES	NO
22.29) Are design and code review coverage measured and recorded?	YES	NO
22.30) Is review efficiency analysed for each project?	YES	NO
22.31) Are code review standards applied?	YES	NO
22.32) Is test coverage measured and recorded for each phase of functional testing?	YES	NO
22.33) Is there a mechanism for assuring the adequacy of regression testing?	YES	NO
22.34) Is a mechanism used for verifying that the samples examined by Software Quality Assurance are truly representative of the work performed?	YES	NO
22.35) Has a managed and controlled process database been established for process metrics data across all projects?	YES	NO
22.36) Is a mechanism used for periodically assessing the software engineering process, and implementing indicated improvements?	YES	NO
22.37) Are analyses of errors conducted to determine their process related causes?	YES	NO
22.38) Is a mechanism used for managing and supporting the introduction of new technologies?	YES	NO

References

- ALLEN, M.J. & YEN, W.M. 2001. Introduction to Measurement Theory. Waveland Pr Inc. 320p.
- AL-TARAWHNEH, M.Y., ABDULLAH, M.S. & ALI, A.B.M. 2011. A proposed Methodology for Establishing Software Process Development Improvement for Small Software Development Firms. *Procedia Computer Science*, (3):893-897p.
- Ang, J., Shaw, N. & Pavri, F. 1995, "Identifying strategic management information systems planning parameters using case studies", *International Journal of Information Management*, 15(6): 463-474.
- ANG, J., SHAW, N. & PAVRI, F. 1995. Identifying strategic management information systems planning parameters using case studies. *International Journal of Information Management*, (15): 463-474p.
- ANON. 2000. The ISO Survey of ISO 9000 and ISO 14000 Certificates – Tenth cycle. <http://www.iso.org/iso/survey10thcycle.pdf> Retrieved: 24 July 2011.
- APRIL, A., HAYAES, J.H., ABRAN, A. & DUMKE, R. 2005. Software maintenance maturity model: The software maintenance process model. *Journal of Software Maintenance and Evolution: Research and Practice*, 17(3):192-223p.
- ARCHER, S. 1988. 'Qualitative' research and the epistemological problems of the management disciplines. (In *Competitiveness and the Management Process* (PETTIGREW, A. Ed). Basil Blackwell, Oxford. p. 265-302).
- ARIAS, G., VILCHES, D., BANCHOFF, C., HARARI, I., HARARI, V. & IULIANO, P. 2012. The 7 key factors to get successful results in the IT Development projects. *Procedia Technology*, 5: 199-207p.
- ASHRAFI, N. 2003. The impact of software process improvement on quality: In theory and practice. *Information and Management*, 40: 677-690p.

AVISON, D.E. & FITZGERALD, G. 2006. Information Systems Development: Methodologies, Techniques and Tools. Berkshire, England: McGraw-Hill Publishing Company. 645p.

BAMFORD, R. & DEIBLER, W. 2003. ISO 9001: 2000 for Software and Systems Providers: An Engineering Approach, 1st Edition. CRC Press. 328p.

BASILI, V.R., CALDIERA, G. & ROMBACH, H.D. 1994. The Goal Question Metric Approach. <ftp://ftp.cs.umd.edu/pub/sel/papers/ggm.pdf>. Retrieved 20 July 2011.

BASILI, V.R., HEIDRUCH, J., LINDVALL, M., MÜNCH, J., SEAMAN, C.B., REGARDIE, M. & TRENDOWICZ, A. 2010. Linking Software and Business Strategy Through Measurement. *Computer*, 43 (4):57-65.

BASKERVILLE, R., TRAVIS, J. & TRUEX, D. 1992. Systems without method: the impact of new technologies on information systems development projects. *IFIP Transactions A*(8):241-269.

BECK, K. & ANDRES, C. 2005. Extreme Programming Explained, 2nd Edition. Addison-Wesley, Boston. 189p.

BEECHAM, S., HALL, T. & RAINER, A. 2003, "Software process improvement problems in twelve software companies: An empirical analysis", *Empirical Software Engineering*, 8(1):7-42.

BELL, J. 1987. *Doing your research project 3rd Ed.* Milton Keynes: Open University Press, London, United Kingdom. 230p.

BENTLEY, C. 2009. PRINCE2: A Practical Handbook 3rd Ed., Butterworth-Heinemann, Oxford. 322p.

BICEGO, A. & KUVAJA, P. 1996, "Software process maturity and certification", *Journal of Systems Architecture*, vol. 42, no. 8 SPEC. ISS., pp. 611-620.

BICEGO, A. & KUVAJA, P. 1996. Software process maturity and certification. *Journal of Systems Architecture*, 42: 611-620p.

BOEHM, B. 2000. Unifying software engineering and systems engineering. *Computer*, (33): 14-116p.

BOLLINGER, T. & McCOWAN C. 1991. A critical look at software capability evaluation. *IEEE Software* 8 (4): 25-41p.

BRACE, I. 2008. Questionnaire Design 2nd Ed. : How to Plan, Structure and Write Survey Material for Effective Market Research (Market Research in Practice). Kogan Page, London, United Kingdom. 304p.

BROWN, H.I. 1977. Perception Theory and Commitment. University of Chicago Press, Chicago, Illinois. 204p.

BROWN, R. 1994. Does America Need ISO 9000? *Machine Design*, 6:70-74p.

BUTLER, K. 1995. The economics benefits of software process improvement. *CrossTalk* (July): 14-17p.

CABALLERO, I., CARO, A., CALERO, C. & PIATTINI, M. 2008, "IQM3: Information quality management maturity model", *Journal of Universal Computer Science*, 14(22): 3658-3685.

CABALLERO, I., CARO, A., CALERO, C. & PIATTINI, M. 2008. IQM3: Information quality management maturity model. *Journal of Universal Computer Science* 14:3658-3685p.

CARNEGIE MELLON SOFTWARE ENGINEERING INSTITUTE. 2008. CMMI® for Development, Version 1.3. p. 29-50.

Carnegie Mellon University. 2009. *Carnegie Mellon Software Engineering Institute: Capability maturity models*. Carnegie Mellon University. Retrieved 29 January 2010 from <http://www.sei.cmu.edu/cmmi/>

CARVALLO, J.P., FRANCH, X. & QUER, C. 2008, "Supporting CMMI Level 2 SAM PA with non-technical features catalogues", *Software Process Improvement and Practice*, vol. 13, no. 2, pp. 171-182.

CERPA, N. & VERNER, J.M. 1998. Case study: The effect of IS maturity on information systems strategic planning. *Information & Management* 34:199-208p.

CERPA, N. & VERNER, J.M. 1998, "Case study: The effect of IS maturity on information systems strategic planning". *Information and Management*, 34,(4):199-208.

CHAN, F.K.Y. & THONG, Y.L. 2009. Acceptance of agile methodologies: A critical review and conceptual framework. *Decision Support Systems*, 46 ::803-814.

CHECKLAND, P. 1981. *Systems Theory Systems Practice*. John Wiley & Sons, Chichester, UK. 330p.

CHENG, C., CHANG, J. & KUO, C. 2011. A CMMI appraisal support system based on a fuzzy quantitative benchmarks mode. *Expert Systems with Applications*, 38:4550-4558.

CHILD, D. 2006. *The Essentials of Factor Analysis* (3rd Edition). Continuum International. 192p.

COAD, P. & YOURDON, E. 1991. *Object Oriented Analysis*, 2nd Ed. Prentice Hall, Englewood Cliffs, New Jersey. 223p.

COHN, T.M & PAUL, R.C. 2001. A comparison of requirements engineering in Extreme Programming (XP) and conventional Software Development Methodologies. *Seventh Americas Conference on Information Systems* (2001). p. 1326-1331.

COLEMAN, G. & O'CONNOR, R. 2008. Investigating software process in practice: A grounded theory perspective. *The Journal of Systems Software*, 81:772-784.

CONGER, S. 1994. *The New Software Engineering*, Wadsworth Publishing Company, Belmont, CA. 800p.

CORTINA, J.M. 1993. What Is Coefficient Alpha? An Examination of Theory and Applications. *Journal of Applied Psychology*, 78(1):98–104.

CRAIG, R.J. 1994. *The No-Nonsense Guide to Achieving ISO 9000 Registration*. New York: Asme Press. 232p.

CRESSWELL, J.W. & CLARKE, V.L.P. 2006. *Designing and Conducting Mixed Methods Research*. Sage Publications, London. 296p.

CROMBIE, A.C. 1996. Commitments and styles of European scientific thinking. *Theoria* 11:65-76.

CURTIS, B. 1992. The Case for Process, in KENDALL K.E. *et al* (eds.), *The impact of computer supported technologies on Information Systems Development*, Elsevier Science Publishers B.V., North Holland, (1992) p. 333-343.

DAVENPORT, T. 1992. *Process Innovation: Reengineering Work Through Information Technology*, Harvard Business School Press, Cambridge, Massachusetts. 352p.

DAYMOND, K.M. 1995. *A guide to the CMM: Understanding the capability maturity model for software*. Annapolis, MD: Process Transition International Inc.

DEMIR, C. & KOCABAŞ, I. 2010. Project Management Maturity Model (PMMM) in educational organizations. *Procedia Social and Behavioral Sciences*, 9:1641-1645.

DIESTE, O. O., JURISTO, N. N., MORENO, A. M., & LOPEZ, M. M. (2000). Integrated Software Engineering and Knowledge Engineering Teaching Experiences. *International Journal of Software Engineering & Knowledge Engineering*, 10(3):275.

DÍAZ-LEY, M., GARCÍA, F., & PIATTINI, M. 2010. *MIS-PyME software measurement capability maturity model – Supporting the definition of software measurement programs and capability determination*. *Advances in Engineering Software*, 41(2010):1223-1237.

DSDM Manual Version 4.2. 2005. DCDM Consortium, Tesseract Publishing, Surrey, UK. 289p.

DURBIN, P.R. 1968. *Logic and Scientific Inquiry*. Bruce Publication Company, University of Wisconsin, Madison. 132p.

EGOROVA, E., TORCHIANO, M. & MORISIO, M. 2010. Actual vs. perceived effect of software engineering practices in the Italian industry. *The Journal of Systems and Software*, 83:1907-1916.

EVA, M. 1994. *SSADM Version 4: A User's Guide*, second edition. McGraw-Hill, Maidenhead, UK. 380p.

EVELEENS, J.L. & VERHOEF, C. 2010. The Rise and Fall of the Chaos Report Figures. *IEE Software* January/February (2010):30-36. Available: ScienceDirect.

FAYAD, M. & LAITINEN, M. 1997. Process assessment considered wasteful. *Communications of the ACM*, 40(11):125-128.

FITZGERALD, B. 1998. Formalized systems development methodologies: a critical perspective. *Information & Management*, 34:317-328.

FLOYD, C. 1981. Process-Oriented Approach to Software Development. Sixth ACM European Regional Conference, 285-294.

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

GANE, C. & SARSON, T. 1979. Structured Systems Analysis: Tools and Techniques. Prentice Hall, Englewood Cliffs, New Jersey. 241p.

GAROUSHI, V. & VARMA, T. 2010. A replicated survey of software testing practices in the Canadian province of Alberta: What has changed from 2004 to 2009? *The Journal of Systems and Software*, 83:2251-2262.

GILLIES, A.C. & SMITH, P. 1994. *Managing software engineering: CASE studies and solutions*. Chapman & Hall, Cornell University. 240p.

GOETHERT, W. 2003. Deriving Enterprise-Based Measures Using the Balanced Scorecard and Goal-Driven Measurement Techniques. Software Engineering Measurement and Analysis Initiative. Carnegie Mellon University. USA. 61p.

GOLDENSON, D. & HERBSLEB, J. 1995. After the Appraisal: A systematic Survey of Process Improvement, its Benefits, and Factors that Influence Success. Software Engineering Institute, Pittsburgh, PA.

GRAY, E. & SMITH, W. 1998. On the limitations of software process assessment and recognition of a required reorientation for global process improvement. *Software Quality Journal*, 7(1):21-34.

GRAY, E.M. & SMITH, W.L. 1998, "On the limitations of software process assessment and the recognition of a required re-orientation for global process improvement", *Software Quality Journal*, 7(1):21-34.

GROSSMAN, J., GROSSMAN, M. & KATS, R. 2006. The First Systems of Weighted Differential and Integral Calculus. Archimedes Foundation, Massachusetts. USA. 66p.

HAIR, J.H. Jr., ANDERSON, R.E., TATHAM, R.L., & BLACK, W.C. 1995. Multivariate Data Analysis, Prentice Hall. 757p.

HARDLE, W. 1990. Applied Nonparametric Regression. Cambridge University Press. 352p.

HARDY, C.J., THOMSON, J.B., & EDWARDS, H.M. 1995. The use, limitations and customization of structured systems development methods in the United Kingdom. *Information and Software Technology*, 37(9):467-477.

HOWARD, G.S., BODNOVICH, T., JANICKI, T., LIEGLE, J., KLEIN, S., ALBERT, P. & CANNON, D. 1999. The efficacy of matching information systems development methodologies with application characteristics - an empirical study. *The Journal of Systems and Software*, 45:177-195.

HOWCROFT, D. & TRAUTH, E. 2004. *The Choice of Critical Information Systems Research*. (In KAPLAN, B., TRUEX, D., WASTELL, D., WOOD-HARPER, A. & DEGROSS, J., eds. Information Systems Research. Springer, Boston. p. 195-211).

http://en.wikipedia.org/wiki/Capability_maturity_model (Retrieved on 15 March 2011)

http://en.wikipedia.org/wiki/Capability_Maturity_Model_Integration (Retrieved on 15 March 2011)

http://en.wikipedia.org/wiki/Process_area_%28CMMI%29 (Retrieved on 15 March 2011)

http://www.iso.org/iso/iso_catalogue/management_standards/iso_9000_iso_14000/iso_9000_essentials.htm (Retrieved on 24 January 2010)

<http://www.praxiom.com/iso-90003.htm> (Retrieved on 24 January 2010)

<http://www.sei.cmu.edu/cmml/> (Retrieved on 24 April 2011)

HUISMAN, H. M. 2000. The Deployment of Systems Development Methodologies: A South African Experience, Ph.D dissertation, Potchefstroom University for CHE, Potchefstroom, South Africa.

HUISMAN, H.M. & IIVARI, J. 2000, "Perceived maturity of IS departments and the deployment of systems development methodologies", Proceedings of the 16th IFIP World Computer Conference, Conference on Software: Theory and Practice, Feng, Y. et al. (Eds.), 21-25 August 2000, Beijing, China, PHEI Publishing House of Electronic Industry, Beijing, China, ISBN 7-5053-6110-4, pp. 135-144

HUISMAN, M. & IIVARI, J. 2006. Deployment of systems development methodologies: Perceptual congruence between IS managers and systems developers. *Information and Management*, 43:29-49.

HYDE, K., & WILSON, D. 2004. Intangible benefits of CMM-based software process improvement. *Software Process: Improvement & Practice*, 9(4): 217-228p.

IBRAHIM, L. & PYSTER, A. 2004. A single model for process improvement. *IT Professional* (6): 43- 49p.

IIVARI, J. & MAANSAARI, J. 1998. The usage of systems development methods: Are we stuck to old practices? *Information and Software Technology* 40(1998):501-510.

IIVARI, J. & HIRSCHHEIM, R. & KLEIN, H.K. 2001. A Dynamic Framework for Classifying Information Systems Development Methodologies and Approaches. *Journal of Management Information Systems* 17(3):179-218.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION.
http://www.iso.org/iso/home/standards/management-standards/iso_9000.htm Accessed 21 July 2011.

JACK, B. & CLARKE, A. 1998. The purpose and use of questionnaires in research. *Prof Nurse*, 14(3):176-179p.

JACKSON, M. 1983. Systems Development, Prentice Hall, Hemel Hempstead, UK. 418p.

JACOBSON, I., BOOCH, G., & RUMBAUGH, J. 1999. The Unified Software Development Process. Addison-Wesley, Boston. 463p.

JAIN, R. & CHANDRASEKARAN, A. 2009. Rapid System Development (RSD) Methodologies: Proposing a Selection Framework. *Engineering Management Journal*, 21(4):30-35.

JEFFRIES, R., ANDERSON, A. & HENDRICKSON, C. 2002. Extreme Programming Installed, Pearson Education, London. 265p.

JIANG, J.J., KLEIN, G. & BALLOUN, J. 1998. Perceptions of system development failures. *Information and Software Technology*, 39:933-937.

JØRGENSEN, M. 2011. Contrasting ideal and realistic conditions as a means to improve judgment-based software development effort estimation. *Information and Software Technology*, 53:1382-1390.

JØRGENSEN, M. & MOLØKKEN-ØSTVOLD, K. 2006. How large are software cost overruns? A judgment-based software development effort estimation. *Information and Software Technology*, 53:1382-1390.

JUNG, H. 2005. Evaluating the ordering of the SPICE capability levels: an empirical study. *Information and Software Technology* (47): 141-149p.

JUNG, H. & GOLDENSON D.R. 2009. Evaluating the relationship between process improvement and schedule deviation in software maintenance. *Information and Software Technology*, 51:351-361.

KEEFE, P. 2004. Oops! Ford and Oracle mega-software project crumbles. <http://adtmag.com/articles/2004/11/01/oops-ford-and-oracle-megasoftware-project-crumbles.aspx>. Accessed: 10 August 2011.

KONRAD, M., CHRISSIS, M., FERGUSON, J., GARCIA, S., HEFLEY, B., KITSON, D., & PAULK, M. 1996. Capability Maturity ModelingSM at the SEI. Software Process. *Improvement & Practice*, 2(1):21-34.

- KRASNER, H. & ZIEHE, T. 1995. Lessons Learned from the Semiconductor Industry for Improving Software Process, Quality, and Reliability. Proceedings of the First World Congress for Software Quality. American Society for Quality Control, San Francisco, CA (Session D): 1-20p.
- KUHN, T. 1977. Second thoughts on paradigms. *The structure of scientific theories*, 2:459-482.
- KUHN, T. 1996. The structure of scientific revolutions 3rd Ed. Chicago: University of Chicago Press. 226p.
- KUILBOER, J.P. & ASHRAFI, N. 2000. Software process and product improvement: an empirical assessment. *Information and Software Technology*, 42:27-32.
- KWAK, Y.H. & IBBS, C.W. 2002. Project Management Process Maturity (PM)² Model. *Journal of Management in Engineering*, July 2002:150-155.
- LAITENBERGER, O. & DEBAUD, J. 2000. An encompassing life cycle centric survey of software inspection. *The Journal of Systems and Software*, 50:5-31.
- LANDRY, P.D. 2001. *The ISO 9000 Essentials: A Practical Handbook for Implementing the ISO 9000 Standards* 3rd Ed. Canadian Standards Association, Canada. 204p.
- LANE, S. & RICHARDSON, I. 2011. Process models for service-based applications: A systematic literature review. *Information and Software Technology*, 53:424-439.
- LEE, C.S. & KO, I.S. & JEONG, C. 2009. Evaluating the Effectiveness of Information Service for SMEs on Information Orientation and Firm Performance. Proceedings of the 42nd Hawaii International Conference on System Sciences (2009). p. 1-9.
- LEEM, C.S. & YOON, Y. 2004, "A maturity model and an evaluation system of software customer satisfaction: The case of software companies in Korea". *Industrial Management and Data Systems*, 104(3):347-354.
- LI ELDON, Y., CHEN, H. & LEE, T. 2002. Software process management of top companies in Taiwan: A comparative study. *Total Quality Management*, 13(5):701-713.

LOKEN, C. & SKRAMSTAD, T. 1995. ISO 9000 certification – Experiences from Europe. American Society for Quality Control, San Francisco, CA (Session Y): 1-11p.

LUKKA, K. 2010. The roles and effects of paradigms in accounting research. *Management Accounting Research*, 21:110-115.

MAGNAYE, R.B., SAUSER, B.J. & RAMIREZ-MARQUEZ, J.E. 2010. System development planning using readiness levels in a cost of development minimization model. *Systems Engineering*, 13:311-323.

MARKIDES, C.C., & WILLIAMSON, P.J. 1994. Related diversification, core competences, and corporate performance. In N. Foss (Ed.). *resources, firms, and strategies: A reader in the resource-based perspective* (1st ed., Vol. 1, pp. 327-344). Oxford, UK: Oxford University Press.

MARSHALL, G. 2005. The purpose, design and administration of a questionnaire for data collection. *Radiography*, 11:131-136.

MARTIN, J. 1989. *Information Engineering*. Prentice Hall, Englewood Cliffs, New Jersey. 178p.

MARTIN, J. 1991. *Rapid Application Development*, Prentice Hall, Englewood Cliffs, New Jersey. 788p.

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.

McLEOD, L., & MACDONELL, S. G. 2011. Factors that Affect Software Systems Development Project Outcomes: A Survey of Research. *ACM Computing Surveys*, 43(4): 24.1-24.56p.

MCMANUS, J. & WOOD-HARPER, T. 2010. A study in project failure. <http://www.bcs.org/content/ConWebDoc/19584>. Accessed: 25 July 2011.

MEILĀ, M. 2007. Comparing Clusterings – an information-based distance. *Journal of Multivariate Analysis*, 98:873-895.

MERRIAM-WEBSTER. 2011. Definition of experiments. <http://www.merriam-webster.com/dictionary/experiments>. Accessed: 20 February 2011.

MERRIAM-WEBSTER. 2011. Definition of paradigm. <http://www.merriam-webster.com/dictionary/paradigm>. Accessed: 20 February 2011.

MERRIAM-WEBSTER. 2011. Definition of survey. <http://www.merriam-webster.com/dictionary/survey>. Accessed: 20 February 2011.

MESQUIDA, A.L., MAS, A., AMENGUAL, E. & CALCO-MANZANO, J.A. 2012. IT Service Management Process Improvement based on ISO/IEC 15504: A systematic review. *Information Technology*, 54:239-247.

MUMFORD, E. 1985. Using Computers for Business, Manchester Business School, Manchester. 70p.

MUMFORD, E. 1995. Effective Requirements Analysis and Systems Design: The ETHICS Method. Macmillan, Basingstoke, UK. 188p.

NA, K., SIMPSON, J.T., LI, X., SINGH, T. & KIM, K. 2007. Software development risk and project performance measurement: Evidence in Korea. *The Journal of Systems and Software* 80 (2007). p.596-605.

NIAZI, M. & ALI BABAR, M. 2009. Identify high perceived value practices of CMMI level 2: An empirical study. *Information and Software Technology*, 51:1231-1243.

NIAZI, M., WILSON, D. & ZOWGHI, D. 2005. A maturity model for the implementation of software process improvement: an empirical study. *The Journal of Systems and Software*, 74:155-172.

NUNALLY, J.C. 2010. Psychometric Theory 3rd Edition. McGraw-Hill, New York. 752p.

OATES, B.J. 2006. *Researching information systems and computing*. Sage Publications, London. 341p.

OLLE, T., HAGELSTEIN, J., MACDONALD, I.G., ROLLAND, C., SOL, H.G., VAN ASSCHE, F.J.M., & VERRIJN-STUART, A. A. 1991. Information System Methodologies: a framework for understanding. Addison-Wesley, University of Michigan. 401p.

OPPENHEIM, A. 2000. A Questionnaire design, interviewing and attitude measurement 3rd Ed. Continuum International Publishing, London, United Kingdom. 312p.

PATEL, C. & RAMACHANDRAN, M. 2009, "Story card Maturity Model (SMM): A process improvement framework for agile requirements engineering practices". *Journal of Software*, 4(5):422-435.

PAULK, M.C., CURTIS, B., CHRISSIS, M.B. & WEBER C.V. 1993. Capability Maturity Model, Version 1.1, *IEEE Software*, July 1993, p. 18-27.

PITTERMAN, B. 2000. Telcordia technologies: the journey to high maturity. *IEEE Software* (July-August): 89-96p.

POPPER, K.R. 1959. *The logic of scientific enquiry*. Hutchinson, London, United Kingdom.

PRIKLADNICKI, R. & AUDY, J.L.N. 2010. *Process models in the practice of distributed software development: A systematic review of the literature*. *Information and Software Technology*, 52:779-791.

QUANG, P.T. & CHARTIER-KASTLER, C. 1991. *Merise in Practice*. Macmillan, Basingstoke, UK. 191p.

RAHIM, M., SEYAL, A.H. & RAHMAN, N. 1998. Use of software systems development methods; An empirical study in Brunei Darussalam. *Information and Software Technology*, 39:949-963.

RAVICHANDRAN, T. T., & RAI, A. 2000. QUALITY MANAGEMENT IN SYSTEMS DEVELOPMENT: AN ORGANIZATIONAL SYSTEM PERSPECTIVE. *MIS Quarterly*, 24(3):381-415.

review of the 1994 CHAOS report. *Information and Software Technology*, 48 (4):297–301.

ROBERTS, T.L., GIBSON, M.L., FIELDS, K.T. & RAINER, R.K. 1998. Factors that impact implementing a systems development methodology, *IEEE Transaction on Software Engineering* 24(8):40–649p.

ROWLANDS, B.H. 2005. Grounded in Practice: Using Interpretive Research to Build Theory. *The Electronic Journal of Business Research Methodology*, 3(1):81-92.

SAEED, G. 2000. Fundamentals of Probability (2nd Edition). Prentice Hall, New Jersey. 438p.

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.

SCHREIBER, G., AKKERMANS, H., ANJEWIERDEN, A., DE HOOG, R., SHADOLT, N., VAN DE VELDE, W., & WIELINGA, B.J. 1999. Knowledge Engineering and Management: The Common KADS Methodology, MIT Press, Cambridge, Massachusetts. 471p.

SCHREIBER, G., WIELINGA, B., & BREUKER, J. 1993. KADS: a principled approach to knowledge-based system development. Academic Press Limited, London. 457p.

SCHWANDT, T.A. 2007. The SAGE Dictionary of Qualitative Inquiry 3rd Ed. Sage Publications, London. 376p.

SEGALSTAD, S.H. 1995. What is needed to comply with ISO 9000, GMP, GLP, and GCP? Laboratory Automation and Information Management 31 (1995). p. 11-24.

SMIDTS, C., HUANG, X. & WIDMAIER, J.C. 2002. Producing reliable software: an experiment, *Journal of Systems and Software*, 61:213-224.

STAHL, B.C. 2008. *The ethical nature of critical research in Information Systems. Information Systems Journal*, 18(2):1-27.

STAHL, B.C. & BROOKE, C. 2008. *Critical IS Research. Communications of the ACM*, 51(3):51-55.

STAPLES, M. & NIAZI, M. 2007. Systematic review of organizational motivations for adopting CMM-based SPI. *Information and Software Technology*, (50):605-620.

STAPLES, M. & NIAZI, M. & JEFFERY, R. & ABRAHAMS, A. & BYATT, P. & MURPHY, R. 2007. An exploratory study of why organizations do not adopt CMMI. *The Journal of systems and Software*, 80:883-895.

STAPLES, M. & NIAZI, M. 2008. Systematic review of organizational motivations for adopting CMM-based SPI. *Information and Software Technology*, 50:605-620.

STAPLES, M., NIAZI, M., JEFFERY, R., ABRAHAMS, A., BYATT, P. & MURPHY, R. 2007. An exploratory study of why organizations do not adopt CMMI. *The Journal of Systems and Software*, 883-895.

STAPLETON, J. 2003. DSDM: Business Focussed Development 2nd Edition, Pearson Education, London. 272p.

STELZER, D., MELLIS, W. & HERZWUM, G. 1996. Software Process Improvement via ISO 9000? Results of Two Surveys among European Software Houses. *Software Process – Improvement and Practice*, 3:197-210.

STELZER, D., MELLIS, W. & HERZWUM, G. 1997. A Critical Look at ISO 9000 for Software Quality Management. *Software Quality Journal*, 6:65-79.

STEVENSON, T.H. & BARNES, F.C. 2001. Fourteen Years of ISO 9000: Impact, Criticisms, Costs and Benefits. *Business Horizons* (May-June): 45-51p.

STEVENSON, T.H. & BARNES, F.C. 2002. What industrial marketers need to know about ISO 9000 Certification – A review, update, and integration with marketing. *Industrial Marketing Management*, 31:659-703.

STOICA, A., BABU, P., & STOICA, P. 2011. Quantitative Framework for Managing Software Life Cycle. *Open Software Engineering Journal*, 5:51-18.

STRAUB, D., GEFEN, D., BOUDREAU, M.C. 2004. The ISWorld Quantitative, Positivist Research Methods Website. <http://dstraub.cis.gsu.edu:88/quant/default.asp> Date of access: 24 March 2011.

SUKHOO, A., BARNARD, A., ELOFF, M.M. & VAN DER POLL, J.A. 2007. An Evolutionary Software Project Management Maturity Model for Mauritius. *Interdisciplinary Journal of Information, Knowledge, and Management*, 2:99-118.

SULAYMAN, M., URQUHART, C., MENDES, E. & SEIDEL, S. 2012. Software process improvement success factors for small and medium Web companies: A qualitative study. *Information and Technology*, 54:479-500.

SUN, Y. & LIU, X. 2010. Business-oriented software process improvement based on CMMI using QFD. *Information and Software Technology*, 52:79-91.

TAYLOR, M.J. & WOOD-HARPER, A.T. 1996. Methodologies and Software Maintenance. *Software maintenance: Research and Practice*, 8:295-308.

TIWANA, A. 2000. The Knowledge Management Toolkit. Prentice Hall, Upper Saddle River, New Jersey. 608p.

TRIBE, J. 2001. Research Paradigms and the Tourism Curriculum. *Journal of Travel Research*, 39:442-448.

VAN DER PIJL, G.J. & SWINKELS, G.J.R & VERRIJDT, J.G. 1997. ISO 9000 versus CMM: Standardization and certification of IS development. *Information and management*, 32:267-274.

VERA, E. 2010. CMMI Certified companies in Mexico and the world. <http://everac99.wordpress.com/2010/10/20/cmmi-certified-companies-in-mexico-and-the-world/> Retrieved: 24 July 2011.

VIDGEN, R., AVISON, D.E., WOOD, R. & WOOD-HARPER, A.T. 2002. Developing Web Information Systems: From Strategy to Implementation, Butterworth-Heinemann, Oxford. 274p.

VON WANGENHEIM, C.G., HAUCK, J.C.R., ZOUCAS, A., SALVIANO, C.F., MCCAFFERY, F. & SHULL, F. 2010, "Creating software process capability/maturity models", *IEEE Software*, 27(4):92-94.

WALSHAM, G. 1995. Interpretive Case Studies in IS Research: Nature and Method. *European Journal of Information Systems*, 4(2):74-81.

WALTERS, S.A., BROADY, J.E. & HARTLY, R.J. 1994. A Review of Information Systems Development Methodologies. *Library Management*, 15(6):15-19.

WARREN, I. 1999. The Renaissance of Legacy Systems: Method Support for Software-System Evaluation, Springer-Verlag, London. 182p.

WELTI, N. 1999. Successful SAP R/3 Implementation. Addison-Wesley, Harlow, UK. 185p.

- WENDLER, R. 2012. The maturity of maturity model research: A systematic mapping study. *Information and Software Technology*, 54:1317-1339.
- WHITTEN, L.W., BENTLEY, L.D. & DITTMAN, K.C. 2001. System Analysis and Design Methods. 5th Ed. New York: McGraw – Hill Irwin. 724p.
- WIELINGA, B.J., STERNER, TH.A., & BREUKER, J.A. 1993. KADS: A modelling approach to knowledge engineering, in B.G. Buchanan and D.C. Wilkinds (eds) *Readings in Knowledge Acquisition and Learning*, Automating the Construction of Expert Systems, Morgan Kaufmann, San Mateo, California. 906p.
- WINTHER, R.G. 2012. Styles, paradigms, and models. *Studies in History and Philosophy of Science*, 43:628-639.
- 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.
- YAMAMURA, G. 1999. Software process satisfied employees. *IEEE Software* (September-October): 83-85p.
- YANG, A., & ZHANG, W. 2009. Based on Quantification Software Quality Assessment Method. *Journal of Software*, 4(10):1110-1118.
- YOURDON Inc. 1993. Yourdon systems Method: Model-Driver Systems Development, Yourdon Press, Eaglewood Cliffs, New Jersey. 1148p.