Universiteit van Pretoria
University of Pretoria

Gordon Institute
of Business Science

The level of adoption and effectiveness of software development methodologies in the software development industry in South Africa.

Name: Vishal Ramnath

Student Number: 29602892

Cell: 0823740660

E-mail: vishal.ramnath@gmail.com

A research project submitted to the Gordon Institute of Business Science, University of Pretoria, in partial fulfilment of the requirements for the degree of Master of Business Administration.

10 November 2010

Abstract

The purpose of this study was to describe the software development industry in South Africa by determining the current and future trends in adopting software development methodologies. The main objective was to determine which factors influence the selection of software development methodologies and whether they are effective in delivering projects successfully within the South African context.

This study found that the Agile method is the most dominant methodology adopted in the software development industry in South Africa and is also the preferred methodology to be used in the future. It is common practice for companies to adopt more than one software development methodology and the Waterfall method is the next methodology most widely used. There is little adoption of CMMI.

All identified factors influence the selection of software development methodologies, however, the distinguishing factors influencing the selection of Agile methods over the Waterfall method in the South African context is team size of between one and five members, project duration of less than three months, iteration length of between two and four weeks and the use of new technology. The most important measurement of project success within the South African context is the delivery of projects on time. The majority of respondents believe that their current dominant methodology is effective in delivering projects successfully.

Keywords

Software Development Methodology; Adoption; Effectiveness

Declaration

I declare that this research project is my own work. It is submitted in partial fulfilment of the requirements for the degree of Master of Business Administration at the Gordon Institute of Business Science, University of Pretoria. It has not been submitted before for any degree or examination in any other University. I further declare that I have obtained the necessary authorisation and consent to carry out this research.

Vishal Ramnath

Date

Acknowledgements

- To my research supervisor Karen Luyt: Your support and guidance throughout this research project has been unwavering. Thank you for your prompt response and advice, even before this topic was defined.

- To GIBS: Thank you for giving me the opportunity to do my MBA. It is truly a privilege to be associated with such an illustrious programme and institute.

- To my fellow MBA colleagues: it has been an unforgettable journey together that will continue with the friendships we have formed.

- To my family and friends: thank you for all your support and encouragement.

- To my darling daughter Siyajna: Sorry for not being there as much as I would have liked to. Thank you for all your patience and understanding even during the most difficult of times. I love you.

- Finally, to my dear wife Surekha: You have been my pillar of strength as always. Thank you for taking on everything in your stride to make the load a lot easier for me. I love you always.

Table of Contents

List of Figures

List of Tables

Acronyms

AIM	Application Implementation Methodology
AM	Agile Modelling
CSSA	Computer Society South Africa
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
DSDM	Dynamic Systems Development Methodology
FDD	Feature Driven Development
GSD	Global Software Development
ISO	International Organisation for Standardisation
ITIL	Information Technology Infrastructure Library
JCSE	Johannesburg Centre for Software Engineering
LD	Lean Development
RUP	Rational Unified Process
SDLC	Software Development Lifecycle
SEI	Software Engineering Institute
SOA	Service Oriented Architecture
SPI	Software Process Improvement
SPICE	Software Process Improvement and Capability Determination, also known as ISO/IEC 15504
XP	Extreme Programming

1. CHAPTER 1: INTRODUCTION TO THE RESEARCH PROBLEM

1.1. Research Title

The level of adoption and effectiveness of software development methodologies in the software development industry in South Africa.

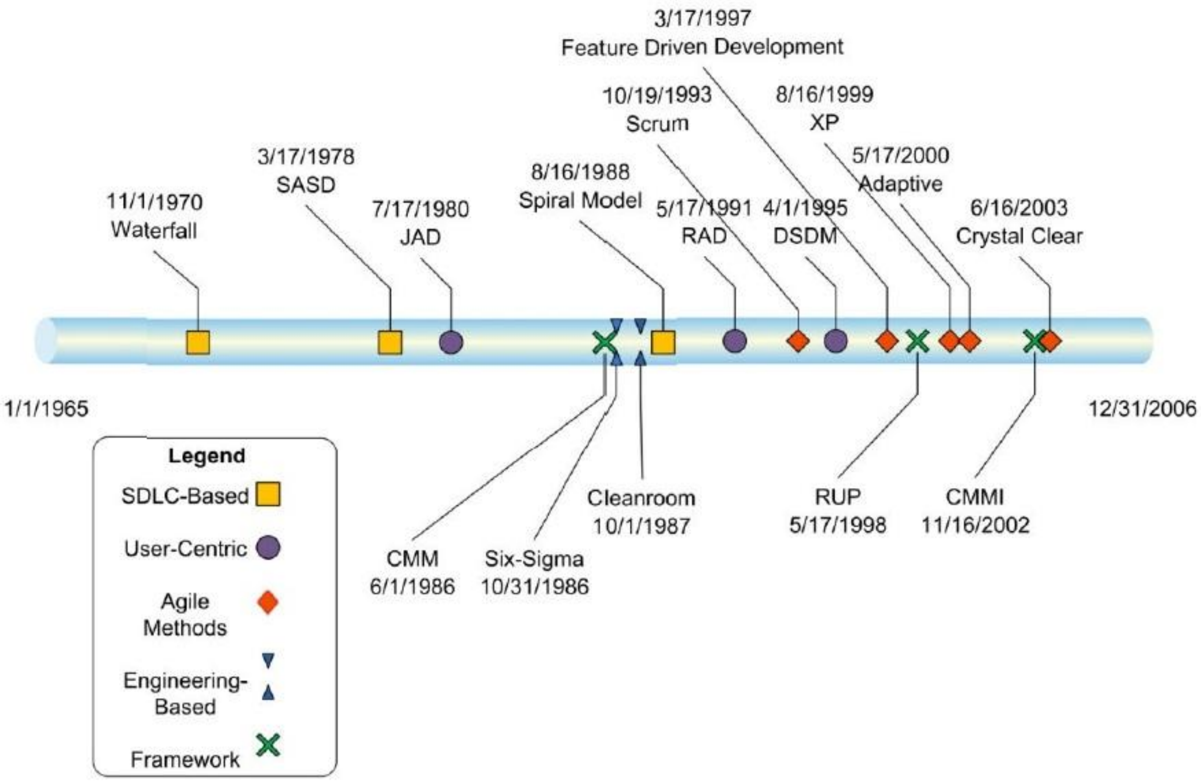
1.2. Research Problem

Now, more than ever, companies want to deliver products and services better, faster, and cheaper while at the same time, in the high-technology environment of the twenty-first century, almost all organisations find themselves building increasingly complex products and services (SEI, 2006). In the modern world of technology, software plays a vital role in running everything from kitchen appliances to aircrafts (Cusick, Prasad & Tepfenhart, 2008).

The Global Software Development (GSD) industry emerged in the wake of the first computers over 60 years ago (Cusick et al., 2008). Since the late 1960s, numerous methodologies have been developed to address the various challenges that occur during software development (Jiang & Eberlein, 2008). Currently, there are maturity models, standards, methodologies, process improvement frameworks and guidelines that can help an organisation improve the way it does business (SEI, 2006).

A timeline of the major software development methodologies have been summarised by Griffin and Brandyberry (2008) and are depicted in Figure 1.1 below. Certain dates are estimated as some ambiguities exist in establishing start dates (Griffin & Brandyberry, 2008). Therefore, software development companies have a range of methodologies which they can implement in order to improve project delivery of increasingly complex products.

Figure 1.1: Software Development Methodology Time



Source: *Griffin and Brandyberry (2008)*

The Waterfall model is still widely in use, especially in smaller software development companies (Paul, 2008). Agile methods are used increasingly to address changing user requirements (Nerur, Mahapatra and Mangalaraj, 2005). The Capability Maturity Model Integration (CMMI) model is being used extensively in the United States (US) and Europe to assist companies in countering risk when awarding tenders (JCSE, 2010).

In order for the local software development industry to become more competitive on a global scale, it will need to fall into line with international standards, so that local companies seeking international contracts will be able to meet the CMMI level specified by international companies (JCSE, 2010).

Despite a range of methodologies available, software project implementations are often over budget, over time and deliver less than the promised benefits (Liu, Chen, Chan & Lie, 2008).

The Chaos Report published by The Standish Group found that only 16% of projects were successful in 1994 (Eveleens & Verhoef, 2008). The Standish Report defines a project as successful if it is completed on time, within budget and with all features and functions initially specified (The Standish Group, 1995). The Standish Group have released seven reports since 1994, the results of which are listed in Table 1.1 below.

Table 1.1: Summary of Results from Chaos Reports

Year	Success	Challenged	Failed
1994	16%	53%	31%
1996	27%	33%	40%
1998	26%	46%	28%
2000	28%	49%	23%
2004	29%	53%	18%
2006	35%	46%	19%
2009	32%	44%	24%

***Source:** Eveleens & Verhoef (2008), Rubinstein (2007), The Standish Group (2009)*

These results show that there has been an improvement in the percentage of successful projects in the last 15 years, however, the percentage of successful projects dropped from 35% in 2006 to 32% in 2009. A more concerning fact in the latest report is the highest percentage of failed projects (24%) this decade.

While there has been a positive shift in the percentage of successful and failed software projects since 1994, the Chaos report does not link this directly to the adoption of software development methodologies. However, after the 2004 report, the chairman of The Standish Group attributed the lower percentage of failed projects to smaller projects using iterative processes as opposed to the Waterfall method.

This has not been quantified and the negative shift in percentages in the last report has not been linked to software development methodologies. Furthermore, there is a concern amongst researchers that the Chaos report is biased towards failure as they do not reveal where their data comes from and the findings from objective research studies do not, in general, support the conclusions of the Standish Group (Glass, 2006).

The only study that has measured the extent to which software development methodologies are used in practise and the future trend in methodology adoption wfas conducted in 1998 by Fitzgerald. This study found that 60% of respondents were not using a formal methodology and 79% of those that were not using a methodology did not intend to adopt one in the future.

1.3. Research Aim and Motivation

Since the study by Fitzgerald in 1998, no further research has been done to evaluate the adoption of software development methodologies and future trends in methodology adoption, even though both CMMI and Agile methods were introduced to the industry during this period. In addition, no such research has been conducted within the software development industry in South Africa.

Given the various types of projects, complexity of products and range of methodologies within the GSD industry, the purpose of this research is to gain a deeper understanding of the state of the software development industry in South Africa. As new and updated software development methodologies are introduced to the industry, an understanding of how companies are responding to these changes will be achieved by determining which software development methodologies have been adopted, factors influencing adoption and the future trends in the usage of methodologies.

The main aim of this study is to determine if the factors that influence the selection of methodologies are realised, thereby enabling companies within the software development industry in South Africa to deliver projects successfully, both locally and internationally.

1.4. Research Objectives

The main objectives of this research are:

- **Objective 1**: to describe the state of the software development industry in South Africa with regards to adoption of software development methodologies.
- **Objective 2**: to determine which factors influence the selection of software development methodologies in the software development industry in South Africa.
- **Objective 3**: to determine if the software development methodologies adopted in the software development industry in South Africa are effective in delivering successful projects both locally and internationally.

1.5. Conclusion

From the above it is clear that there isn't a single methodology that caters for all types of projects, product complexities and development sourcing options within the GSD industry. The software development industry in South Africa has not been researched and could therefore benefit from understanding what methodologies are being used, the reasons they are used and determining if they are able to deliver projects successfully both locally and internationally.

The contents of this report are divided into chapters as follows:

- Chapter two analyses the constructs and links these too previous research on software development methodologies to derive the questions that were included in a survey targeted at the software development industry in South Africa.
- Chapter three defines the research questions that must be answered in order to meet the research objectives of this study.
- Chapter four defines the research methodology used to sample, collect and analyse the data required to answer the research questions.
- Chapter five presents the results from the sample by analysing the data collected in terms of the three research questions.
- Chapter six discusses the results in terms of the data, research questions, literature reviewed and the research topic.
- Chapter seven is the concluding chapter that highlights the main research findings, limitations of this study and recommendations for future research.

2. CHAPTER 2: LITERATURE REVIEW

2.1. Introduction

The theory reviewed in this chapter is divided into three main areas, namely, software development methodologies, global software development and successful software development projects. By linking these three areas, this study aims to meet the research objectives listed in section 1.4 above.

Firstly, a summary of the various software development methodologies is provided in order to explain why and how methodologies have evolved. The methodologies which are still widely in use today were identified to determine which methodologies are relevant to include in this study. Thereafter, the main benefits and shortcomings of the selected methodologies were studied to determine the factors influencing methodology selection so that this study can explore whether these are realised within the software development industry in South Africa. The summary of this section will link the various literature and debates regarding software development methodologies.

Secondly, global software development was researched to define the environment in which South African companies can compete both locally and internationally. This will determine if the nature of the environment in which the organisation competes is a factor that influences the selection of methodologies.

Lastly, the criterion for measuring successful software development projects is defined. These measurements contribute to the factors that influence methodology selection but more importantly are used to determine if the reasons for selecting methodologies result in the delivery of successful projects.

2.2. Software Development Methodologies

Problems occur in software development mainly due to the difficulty of establishing and stabilising the requirements, the changeability of the software and interactive dependency of software, hardware and human beings (Pikkarainen, 2008). Since the late 1960s, numerous methodologies have been developed to address the various challenges that occur during software development (Jiang & Eberlein, 2008).

Methodologies impose a disciplined process upon software development with the aim of making software development more predictable and more efficient (Awad, 2005). Theory advocates three categories which each methodology can be grouped into based on the underlying principles of the methodology (Jiang & Eberlein, 2008; Paul, 2008; Pikkarainen, 2008; Vinekar, Slinkman & Nerur, 2006). These three categories are traditional methodologies, Agile methodologies and Software Process Improvement (SPI) frameworks.

2.2.1. Traditional Methodologies

Traditional methodologies are plan driven, requiring software development projects to begin with the elicitation and documentation of a complete set of requirements, followed by an architectural design, development and inspection (Awad, 2005). This sequential series of steps are the underlying principles of traditional (or SDLC-based) development models such as the Waterfall model, unified process and spiral model (Awad, 2005; Paul, 2008).

In the study by Paul (2008), the Waterfall model is identified as the traditional methodology still widely in use, especially in smaller software development companies as it is easy to understand and adopt. This implies that the size of the company and effort of implementation are factors influencing methodology selection.

Traditional development models assert that complex systems can be built in a single pass without revisiting requirements or design ideas in light of changing business needs (Szalvay, 2008). Therefore, perfect knowledge of the client's needs are required upfront (Paul, 2008), which is not the case in the complex and high-technology environment of the twenty-first century (SEI, 2006). Therefore, another factor influencing methodology selection is the complexity of requirements, identified as a shortcoming of the Waterfall model.

The spiral model has been refined further and its basic principles are currently essential parts of the Rational Unified Process (RUP), Extreme Programming, and generally the Agile software development framework (Paul, 2008). These iterative development methods were developed in response to the weaknesses of the classic Waterfall model (Paul, 2008).

Since the spiral model and unified process has evolved and are no longer widely used, these models are excluded from the scope of this study. However, the Waterfall model is still widely in use (Paul, 2008; Vinekar et al., 2006) and it is therefore applicable to assess the adoption of this model within the context of the software development industry in South Africa. Therefore, from the traditional methodologies, the Waterfall model is the only model selected in this study.

2.2.2. Agile Methodologies

Agile development is not a formally specified software development process model but rather a set of multiple development methods implementing agile practices and values (Paul, 2008). These well defined development methods, which are listed in Table 2.1 below, exhibit the following values documented in the Manifesto for Agile Software Development:

- *Individuals and interactions* over processes and tools
- *Working software* over comprehensive documentation
- *Customer collaboration* over contract negotiation
- *Responding to change* over following a plan

Cohen et al. (2004) defines four prescriptive characteristics of Agile methods which govern which Agile method to use. These characteristics, regarded as factors that influence methodology selection, include the following:

- **Team size**: number of developers involved in the project.
- **Iteration length**: duration of each iteration.
- **Distributed support**: support for distributed teams.
- **System criticality**: applications that may endanger human life, like manned space missions must undergo much stricter quality control than less critical applications.

The prescriptive characteristics for each Agile method is summarized in Table 2.1 below and can be used to select the most appropriate Agile method within a set of conditions.

Table 2.1: Prescriptive Characteristics of Agile Methods

Agile Methods	Team Size	Iteration Length	Distributed Support	System Criticality
Extreme Programming (XP)	2-10	2 weeks	No	Adaptable
Scrum	1-7	4 weeks	Adaptable	Adaptable
Crystal Methods	Variable	< 4 months	Yes	All types
Feature Driven Development (FDD)	Variable	< 2 weeks	Adaptable	Adaptable
Lean Development (LD)	Not applicable			
Dynamic Systems Development Methodology (DSDM)				
Agile Modelling (AM)				

Source: *Cohen et al. (2004)*

Given that Agile methodologies are a set of several development methods implementing agile practices (Paul, 2008) and these Agile methods are well defined together with the prescriptive characteristics for selecting Agile methods (Cohen et al., 2004), little value can be derived by describing or studying each Agile method in detail. Instead, defining the underlying agile principles and the benefits and limitations of Agile development will contribute to determining what other factors influence the selection of Agile methodologies, apart from the prescriptive characteristics defined by Cohen et al. (2004).

One of the principles of Agile development is that the most efficient and effective method of conveying information to and within the development team is through face-to-face conversation rather than talking through documents (Williams, 2010). Agile methodologies focus on producing minimal documentation that is sufficient to support the development of working software and to satisfy the customers' explicit needs for documentation (William, 2010).

A U.S. Department of Defence study illustrates that 45% of software features fail to meet user needs and requirements, and Agile software development approaches are proposed as solutions to improve a software team's ability to embrace and respond to changing requirements (Lee & Xia, 2010). A principle of Agile development is to welcome changing requirements, even late in development (Williams, 2010). Therefore, changing requirements is a factor that influences methodology selection.

Nerur et al. (2005) argue that Agile methodologies are ideal for projects that exhibit high variability in tasks (due to changing requirements), in the capabilities of people, and in the technology being used. Therefore, resource skills and technology are factors that influence methodology selection.

2.2.3. Software Process Improvement Frameworks

Software Process Improvement (SPI) has been a long-standing approach promoted by software engineering researchers, intended to help organisations develop higher-quality software more efficiently (Staples et al., 2007). Process capability maturity models such as Capability Maturity Model (CMM), Capability Maturity Model Integration (CMMI) and ISO/IEC 15504 (SPICE) are SPI frameworks for defining and measuring processes and practices that can be used by software development companies (Staples et al., 2007).

CMMI is the integration of several models including elements of both CMM and SPICE and is used to assist an organisation to achieve a level where continuous, optimised improvement of software development is possible (Pikkarainen, 2008). In addition, The CMMI model is being used extensively in the US and Europe (JCSE, 2010), and is therefore the only model from the SPI framework that is included in the scope of this study. In addition, Paul (2008) briefly introduces Information Technology Infrastructure Library (ITIL) and

International Organisation for Standardisation (ISO) 9001 but defines them as process models not strictly related to software development, therefore, these models are excluded from this study.

The two key challenges of CMMI is the overly heavy and time-consuming assessments and the risk that the achievement of CMMI levels forces the developers to spend more time writing documents than implementing software products (Pikkarainen, 2008). Niazi and Babar (2009) argue that small and medium size companies should not be seen at fault for not adopting CMMI and suggest that the model should be tailored for these companies. Therefore, company size, time and resource constraints are all factors that influence methodology selection.

2.2.4. Summary of Software Development Methodologies

The introduction of Agile methodologies as another answer to some of the problems encountered during software development has caused a heated debate amongst software developers since early 2000 (Jiang & Eberlein, 2008). On one hand, software developers dismiss Agile methodologies and strongly support the value of the traditional methodologies, while others insist that Agile methodologies will replace traditional models and apply to all software projects sparking a heated debate sometimes referred to as the “Methodology War” (Jiang & Eberlein, 2008).

In a study to determine if Agile and traditional methods can coexist, Vinekar et al. (2006) found an increased acceptance of Agile methods in traditional software development organisations, however, the majority of these organisations prefer to sustain both forms of development. This implies that this study should look for the adoption of a dominant methodology as well as a secondary methodology within companies, as value can be derived from understanding if and why more than one methodology is used.

The initial adoption of a software development methodology is normally an organisational decision (Khalifa & Verner, 2000). Often, however, multiple methods are adopted and used together for the same project, but at different phases of the development life cycle (Khalifa & Verner, 2000). It can be argued further that companies have multiple teams and run multiple projects each with a varying degree of complexity to which a different methodology may be suitable. This is supported by Vinekar et al. (2006) who found that the majority of organisations prefer to sustain both the Agile and traditional methods. This advocates that the unit of analysis should be at the team level and not at a company level.

While the “Methodology War” still continues, the Introduction of the CMMI model provides a framework made up of best-of-the-best processes gleaned from multiple disciplines (Siviy, Penn & Harper, 2005). However, in their study Siviy et al. (2005) conclude that CMMI and Six Sigma are different types of models, therefore, one model cannot subsume the other. Furthermore, Agile development methods and CMMI best practices are often perceived to be at odds with each other (Glazer, Dalton, Anderson, Anderson, Konrad & Shrum, 2008).

Glazer et al. (2008) conclude by listing the differences between Agile methods and CMMI. CMMI is more suited to large organisations, is resource intensive and requires heavy documentation, while Agile methods are more suited to small and medium organisations, promotes minimal documentation and requires skilled resources to code and interact face to face rather than through documentation.

The various software development methodologies in use today have their benefits and limitations. The literature review has shown that CMMI and Agile methods are the most widely adopted methodologies in software development. While these and other new models are improvements on the limitations of previous methodologies, no methodology fully subsumes the rest, resulting in the traditional Waterfall model still being used today.

In their 2008 study, Griffin and Brandyberry (2008) conclude by pointing to the lack of good systematic research related to the usage of software development methodologies and suggest the relatively simple question of “who is using what, for what, and why?” needs to be answered before questions that are even more meaningful to practice can be posed.

2.3. Global Software Development

Computers were born as an international industry some 60 years ago and today it is a global business which is growing in size, scope, and geography (Cusick et al., 2008). Increasingly, this global industry also produces its products using globally dispersed and culturally diverse teams (Cusick et al., 2008). In this global industry, a number of development sourcing options exist, each with its own advantages and disadvantages as depicted in Table 2.2 below. The model and parties involved in a typical outsourced software development project is depicted in Appendix 2.

Table 2.2: Development Sourcing Options

Development Options	Advantages	Disadvantages
In-house development	More control	Higher cost in developed countries
On-site contractors	Easy for scaling up or down	Higher risk
Onshore outsourcing	Leverage skills to do some work offsite at lower cost	
Near Shore outsourcing (aligned time zone proximity)	Cost advantage	Communication and coordination challenges
Far Shore outsourcing (opposite time zones)	Cost advantage	Communication and coordination challenges
In-house off shoring	Removes middle man and less risky	Additional overhead
Hybrid sourcing (two or more approaches above)	As above	As above

Source: *Cusick et al. (2008)*

Well before 1993, companies began shifting software development to offshore sites in countries like Ireland and India (Ravichandran & Ahmed, 1993). The main benefit to offshoring is lower cost. However, other benefits include the ability to undertake large projects, to expand markets and to achieve agility in development and operation (Batra, 2009).

Companies that operate within the software development industry in South Africa can choose their market segment from the various development sourcing options summarised in Table 2.2 above. These development options are regarded as factors that influence the selection of methodologies. For example, in the case of outsourcing where communication and coordination is a challenge, Agile methods might not be suitable given the focus on short iterations and the need for face-to-face interaction. This is supported by Batra (2009) who provides evidence that agile practices will need to be modified for outsourced software projects.

This study first aims at describing the South African software development industry in the GSD environment by determining which development sourcing options are being utilised. The main aim is to determine if the adopted software development methodologies effectively support the development options exercised by companies within the software development industry in South Africa. This will determine if these companies are able to successfully deliver projects both locally and internationally.

2.4. Successful Software Development Projects

Agarwal and Rathod (2005) define cost, time and scope as the success indicators of a project. This is supported by the Chaos report which defines a project as successful if it is completed on time, within budget and with all features and functions initially specified (The Standish Group, 1995).

However, software development typically requires software development teams to make trade-off decisions among these interdependent and conflicting goals of time, cost, and scope (Lee & Xia, 2010). Despite these conflicts, the performance of project managers is usually evaluated by these triple golden criteria of staying within budgeted cost, keeping to the schedule, and meeting requirements (Liu et al., 2008). Therefore, these three factors will be used to measure the success of software development projects in this study.

Agarwal and Rathod (2005) found that scope is considered to be the most important factor for project success for both internal and external stakeholders. Therefore, while all three measurements are used, their importance may differ amongst development teams and customers resulting in a different weighting or priority of the three measurements.

Since these measures require trade-offs, have different priorities and are used to evaluate the performance of managers, they are therefore included as factors that influence the selection of methodologies. Given that scope is the most important measurement and Agile methodologies are known for managing scope and complex requirements (Lee & Xia, 2010; Williams, 2010), it is understandable for individuals to select a methodology that has the most influence on their measurements.

2.5. Literature Conclusion

The JCSE (2010) argues that the local software development industry needs to fall into line with international standards. The CMMI model is being used extensively in the US and Europe (JCSE, 2010), however, the literature review shows a wide adoption of other methodologies, the Agile method in particular. A dominant methodology in the GSD industry is nonexistent and if South African companies are not competing internationally, then there is no motivation to fall into line with international standards, contrary to the view of the JCSE (2010).

In their 2008 study, Griffin and Brandyberry (2008) conclude by pointing to the lack of good systematic research related to the usage of software development methodologies. A more recent study by Lee and Xia (2010) focuses only on Agile methodologies as the answer to the changing requirements, despite the fact that theory illustrates the need for the Waterfall method in practice (Paul, 2008) and the fact that Agile methods are not suited to outsourced projects (Batra, 2009). The relevance of this study is further highlighted in the conclusion by Griffin and Brandyberry (2008) who suggest the relatively simple question of “who is using what, for what, and why?” needs to be answered.

2.6. Conclusion

From the three categories which software development methodologies can be grouped into based on their underlying principles, the literature review presents three methodologies still widely in use. These methodologies which have been included in the scope of this study are the Waterfall model, Agile methods and CMMI.

The reasons for adoption and the limitations of these software development methodologies have been highlight throughout the literature and are considered as factors influencing the selection of methodologies. These factors are summarised in Table 2.3 below and have been incorporated into the design of the research questionnaire.

Table 2.3: Literature Summary

Factors Influencing Selection	Literature
Size of company	Niazi & Babar, 2009; Paul, 2008
Effort of implementation	Niazi & Babar, 2009; Paul, 2008
Complexity of requirements	Paul, 2008; SEI, 2006; Szalvay, 2008
Team size	Cohen et al., 2004
Iteration length	Cohen et al., 2004
Distributed support	Cohen et al., 2004
System criticality	Cohen et al., 2004
Changing requirements	Williams, 2010; Lee & Xia, 2010
Resource skills	Nerur et al., 2005; Niazi & Babar, 2009
Technology	Nerur et al., 2005
Development sourcing options	Batra, 2009; Cusick et al, 2008; Ravichandran & Ahmed, 1993
Measurements of project success • Time • Budget • Scope	Agarwal & Rathod, 2005; Eveleens & Verhoef, 2008; Lee & Xia, 2010; Liu et al., 2008; Rubinstein, 2007; The Standish Group, 1995; Williams, 2010

3. CHAPTER 3: RESEARCH QUESTIONS

The literature review does not provide a silver bullet for software development companies to adopt, but suggests that each model is more suitable under certain conditions. These conditions can be influenced by the company, market, product, type of project or the principles of the methodology itself.

None of the previous studies have tested these conditions across all methodologies within a single country. In addition, no research has been conducted with regards to the adoption of software development methodologies within the South African software development industry.

Therefore, this study investigated the level of adoption and effectiveness of software development methodologies in the South African software development industry by answering the following research questions:

- **Question 1**: what are the current and future trends in adopting software development methodologies in the South African software development industry?
- **Question 2**: what are the factors influencing the selection of software development methodologies in the South African software development industry?
- **Question 3**: are the software development methodologies adopted by companies in the South African software development industry effective in delivering successful projects locally and internationally?

4. CHAPTER 4: RESEARCH METHODOLOGY

4.1. Introduction

This chapter defines the research methodology used to conduct this study. The population, unit of analysis and sample are described, followed by the method used to collect and analyse the data.

4.2. Research Design

The design of this empirical study is quantitative in nature. Descriptive research was used to answer all three research questions. The rationale for this method is that significant research has been conducted in a single software development methodology (Paul, 2008; Salo & Abrahamsson, 2008; SEI, 2006), within software start-ups (Coleman & O’Connor, 2008) or comparisons between certain software development methodologies (Glazer et al., 2008; Siviy et al., 2005).

The research by Fitzgerald (1998) studied the extent to which software development methodologies are used in practice and the future trend in software development methodology adoption. However this study was conducted 12 years ago, before CMMI and Agile methods were introduced. In addition, this study measured if a formal software development methodology is adopted or not, and does not analyse which software development methodologies are used.

Griffin and Brandyberry (2008) conclude by suggesting that the relatively simple question of “who is using what, for what, and why?” needs to be answered.

Zikmund (2003) states that descriptive research seeks to determine the answers to who, what, when, and how questions.

Exploratory research was not considered as no new concepts are required. This study builds on past research by combining previous studies in order to describe the software development industry in South Africa, across the software development methodologies most widely used today.

The 16 major software development methodologies (Figure 1.1) defined by Griffin and Brandyberry (2008) were reviewed in this study. After an extensive literature review, only the software development methodologies still widely in use were included in this study. These are the Waterfall Model, Agile Methodologies and CMMI which are supported by Cohen et al. (2004), JCSE (2010), Lee and Xia (2010), Paul (2008), Pikkarainen (2008), and SEI (2006).

The factors that influence methodology selection were derived and highlighted throughout the literature review. These factors formed the basis of the closed questions and the response options available to the respondent on the questionnaire form. In addition, survey questions from past studies provided input into the questionnaire design, research questions and response options.

4.3. Population

The population for this study included all software development companies based in South Africa that develop software solutions for the local and international market as well as for in-house use. This includes both listed and unlisted companies. In the case where companies have international branches, this research is only applicable to the operations based in South Africa.

Since the context of this study is the software development industry in South Africa, all companies that compete within this industry were sampled, including international companies such as Indian companies that have established an office in South Africa.

4.4. Unit of Analysis

In software companies, technical survival and success can depend heavily on the managers and executives who are responsible for technical strategies (Coleman & O'Connor, 2008). Therefore, as software development methodologies evolve these managers must decide whether to adapt to these changes. This requires both time and money which start-ups and small companies may not have.

When considering changes in software development methodologies, previous software process experience is often considered an indicator of success (Coleman & O'Connor, 2008). By contrast, previous negative experience of software process improvement can act as a de-motivator for practitioners towards implementing change (Coleman & O'Connor, 2008).

Therefore, the unit of analysis is at the team level and the questionnaire was targeted at Software Development Managers or managers of software development teams to be completed for the team he or she manages. The company was not selected as the unit of analysis as many software development companies have a number of development teams. Teams within the same company may adopt different methodologies based on different factors and operating environments, therefore, the unit of analysis will be at the team level.

4.5. Sampling Method

Two types of non-probability sampling were used as the sampling method. Judgement sampling was used as the first sampling technique based on the researchers experience and knowledge of software development experts, companies and institutes. Snowballing sampling was used as the second sampling technique.

4.5.1. Judgement Sampling

Judgement sampling was used to contact the JCSE in order to obtain their database of software development companies. Since the JCSE focuses on CMMI and Agile methods, contacting this database of companies only, may introduce response bias towards these methodologies. Therefore, software development companies listed in the Information Technology (IT) directory on the ITWeb website were also contacted. These two lists of companies were then combined. This combined list of companies was contacted via electronic mail (e-mail) to identify the appropriate individual(s) within the company to complete the questionnaire.

A second form of judgement sampling was used whereby the contacts of the researcher were surveyed. For these samples, the individuals within the companies were contacted directly via e-mail.

4.5.2. Snowball Sampling

The snowball sampling technique was applied to the initial set of judgement samples in order to obtain referrals and to get a better representation of the population. This was achieved through a referral section at the end of the questionnaire (section 6 of questionnaire) which provided the respondent with the option to refer other individuals who are in a position to complete the questionnaire.

4.6. Sample Size

At least 30 responses were expected with the majority of the responses coming from the researchers contacts. This sample size further supports the selection of descriptive trend analysis because the sample and sub groups will not be large enough to do statistical analysis for each methodology and factors influencing selection.

4.7. Data Collection

An electronic, self administered survey consisting of structured questions was used to collect the necessary data. SurveyMonkey (2010), an online questionnaire tool, was used as the data collection instrument. The questionnaire was made up of simple-dichotomy questions, determinant-choice questions, checklist questions and filter questions (Zikmund, 2003).

In SurveyMonkey (2010), radio buttons were used to limit the user to a single response when a single selection was required for simple-dichotomy and determinant-choice questions. In the case of checklist questions, checkboxes were used to allow the user to select multiple responses. When ranking and percentage responses were required such as in question 2.2 (refer to questionnaire in Appendix 2), numerical textboxes were used with data validation to ensure that the correct total was reached.

Page logic was used in SurveyMonkey (2010) to build filter questions whereby the respondent was directed to a page based on the response to the previous question. This allowed questions to be filtered out if they were not applicable.

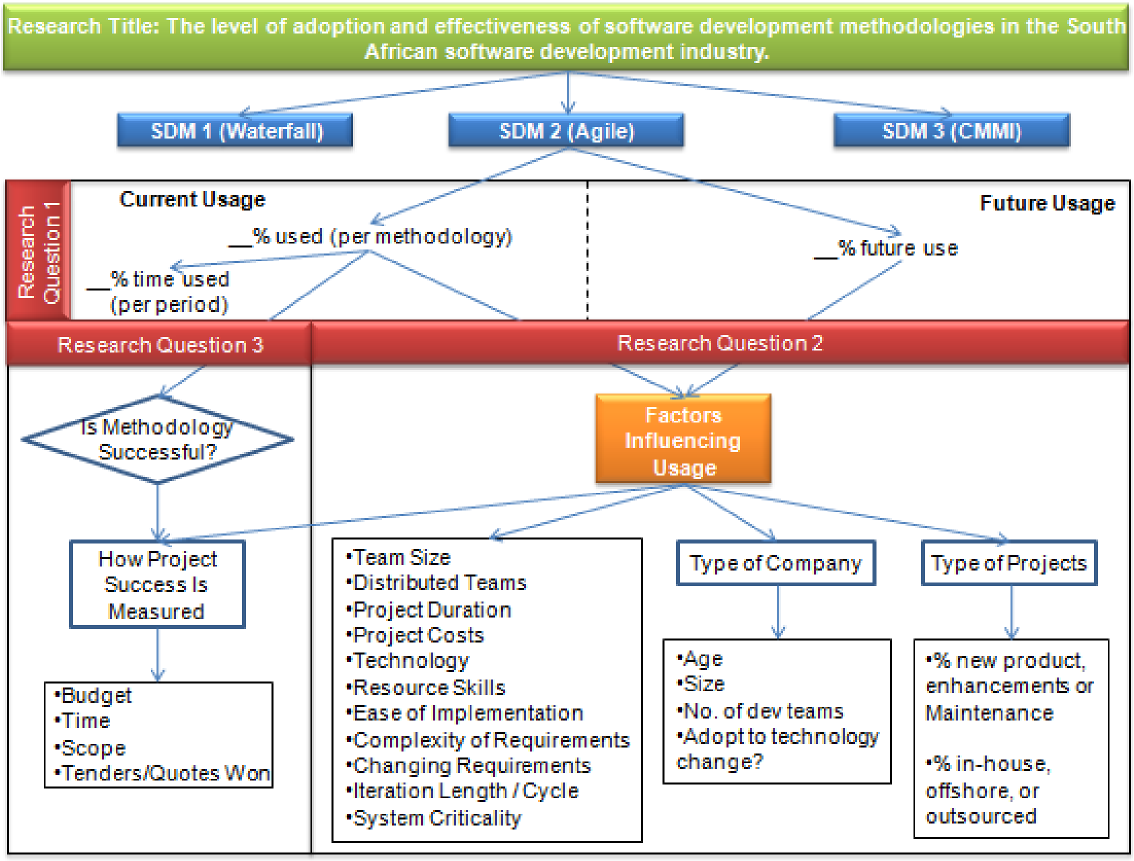
For example, if the response to question 4.1 is none, the next page containing the rest of the questions for section 4 were skipped as these are not applicable.

The above restrictions and validation to responses on the input screen were implemented to reduce errors and ensure that the data is captured accurately and consistent across all respondents. The survey went live on 9 August 2010 and was open for a period of four weeks until 7 September 2010. A reminder was sent after three weeks to all respondents who had not confirmed via e-mail when they had completed the survey. After the survey was closed, the raw data was downloaded to Microsoft Excel.

4.8. Data Analysis

The raw data was cleaned up and analysed in Microsoft Excel using trend analysis to answer all three research questions. A graphical representation of how the data was analysed is provided in Figure 4.1 below.

Figure 4.1: Graphical Representation of Research Methodology


Research Title: The level of adoption and effectiveness of software development methodologies in the South African software development industry.
SDM 1 (Waterfall)
SDM 2 (Agile)
SDM 3 (CMMI)
Research Question 1
Current Usage
__% used (per methodology)
__% time used (per period)
Future Usage
__% future use
Research Question 3
Research Question 2
Is Methodology Successful?
Factors Influencing Usage
How Project Success Is Measured
•Budget
•Time
•Scope
•Tenders/Quotes Won
•Team Size
•Distributed Teams
•Project Duration
•Project Costs
•Technology
•Resource Skills
•Ease of Implementation
•Complexity of Requirements
•Changing Requirements
•Iteration Length / Cycle
•System Criticality
Type of Company
•Age
•Size
•No. of dev teams
•Adopt to technology change?
Type of Projects
•% new product, enhancements or Maintenance
•% in-house, offshore, or outsourced

Trend analysis was used to interpret the data and identify patterns, trends and the most frequent responses which are presented in pie charts and bar charts. This is similar to the analysis performed by Fitzgerald (1998) and the presentation of results in the Chaos Report (The Standish Group, 1995). The analysis for each research question as depicted in Figure 4.1 is summarised below.

In answering the first research question, descriptive analysis was used to list the percentage of respondents using each software development methodology currently, as well as the percentage of respondents planning to use the methodology in the future.

To answer the second research question, the data is presented in a frequency table where the frequency and percentage of responses are calculated for each factor influencing the selection of software development methodologies. Zikmund (2003) states that this provides the most basic form of information and in many cases, the most useful information.

However, to extract full value from the data (Zikmund, 2010) and link research question one and question two, cross-tabulation was used to draw comparisons between the most frequent methodologies used and the most frequent factors influencing methodology selection.

In answering the third research question, a frequency table was used to measure the most common methods of measuring project success within the software development industry in South Africa. In addition, cross-tabulation was used to determine which methodologies deliver more successful projects by comparing successful projects and methodology used.

4.9. Limitations

This study is limited to the condensed list of widely used software development methodologies supported by the literature review. Any other methodologies are grouped into a category named other.

This study did not use statistical analysis to answer the research questions because discrete data was collected. However, this study illustrates the current and future trends in the adoption of software development methodologies which will allow future researchers to statistically test the software development methodologies and factors identified as most frequent in this study.

A further limitation is that duplicate responses may be received from managers within the same company as the questionnaire does not require the respondent to enter their name or company name. The rationale for this is that the sample is small and to improve the response rate, confidentiality must be ensured so that individuals are willing to provide information they may regard as sensitive such as the type of projects and success of projects within the company.

To minimise this duplication, an explanation regarding which person in the organisation should complete the survey was given at the beginning of the questionnaire. The researcher also tracked the number of companies and individuals contacted as well as the number of responses received. If duplications are received when more than one manager responds for a team, the information will be the same as the team will operate in the same way.

4.10. Conclusion

This chapter described the research methodology used in this study. The literature and Zikmund (2003) supported the use of descriptive statistics to answer all three research questions. The population of the study is all software development companies in the software development industry in South Africa. The unit of analysis is at the team level and both judgement and snowball sampling techniques were used. The data was collected using SurveyMonkey (2010), an online survey tool. The data was analysed using trend analysis as discrete data was collected.

5. CHAPTER 5: RESULTS

5.1. Introduction

This chapter discusses the sample data collected from the online questionnaire and then applies the research method defined in chapter four to analyse the sample data and answer the three research questions defined in chapter three.

The details of the sample are first discussed, followed by an overview of how the data was assessed. Lastly, the results are presented according to the research methodology under the two main sections, current trends and future trends, in the adoption of software development methodologies.

5.2. Sample

The JCSE was unwilling to share their database of companies as it is against their policy. However, six software development companies publicly listed as partners on the JCSE website formed the first judgement sample. In addition, a total of 75 companies listed under the software development category on the ITWeb website formed the second judgement sample.

This combined list of 81 companies was contacted via e-mail to determine the number of development teams and the contact details of individuals in each company who were in a position to complete the survey. This judgement sampling method resulted in responses by six individuals in four companies to which the survey link was sent via e-mail.

The third judgement sample consisted of 25 IT professionals who are contacts of the researcher. An e-mail was sent to these individuals requesting the names of companies and individuals to complete the survey. This sampling technique produced 11 responses whereby the contact details of 35 individuals in 18 companies were obtained.

The snowball sampling technique produced six referrals from four respondents. The link to the survey was e-mailed to five individuals only, as the last referral was received shortly before the survey closed.

In summary, an e-mail was sent to 111 companies requesting the contact details of managers of software development teams that were suitable to complete the survey. The link to the online survey was then sent to 46 team managers from 22 companies. A total of 32 team managers responded resulting in a team response rate of 29% for companies initially contacted and a response rate of 69% for individuals that the survey link was sent to.

5.3. Data

The raw data was downloaded to Microsoft Excel. The spreadsheet was then cleaned up by deleting the blank rows and columns. The data was checked for invalid or non responses and none of the 32 responses were discarded as all responses were complete.

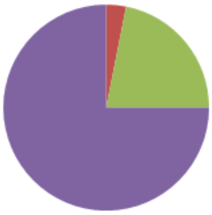
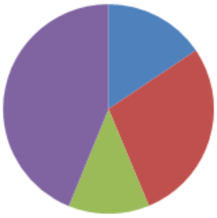
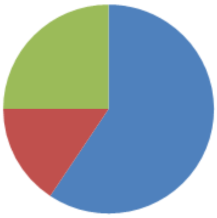
5.3.1. Companies

A summary of the company data is displayed in pie charts in Figure 5.1 below. The majority of companies (75%) that responded to the survey have been operating for more than ten years with only one company in existence for less than five years.

The data also shows that 44% of the companies have more than 500 employees, while 59% of the respondents regard their company as market leaders. None of the respondents believe that their company remains static.

The profile of respondents indicates that the majority of software development companies have been in operation during the introduction of CMMI and Agile methods. In addition, the majority of companies are large organisations and regard themselves as market leaders.

Figure 5.1: Company Data

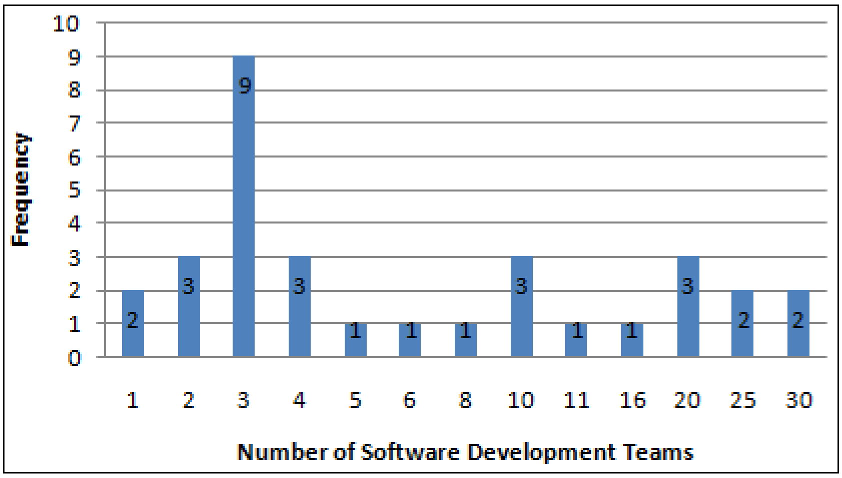
Age of company.		**Answer Options**	**Percent**	**Count**
		< 2 years	0%	0
		2-5 years	3%	1
		5-10 years	22%	7
		> 10 years	75%	24
Number of employees.		**Answer Options**	**Percent**	**Count**
		< 50 employees	16%	5
		50-200 employees	28%	9
		201-500 employees	13%	4
		> 500 employees	44%	14
Adopting new technologies and methods.		**Answer Options**	**Percent**	**Count**
		Market Leader	59%	19
		Market Follower	16%	5
		Conservative	25%	8
		Static	0%	0

The holding company of 88% of respondents is South African. This distribution suggests that the findings will be heavily skewed towards local companies. However, six of these companies operate internationally by engaging in offshore development projects.

5.3.2. Software Development Teams

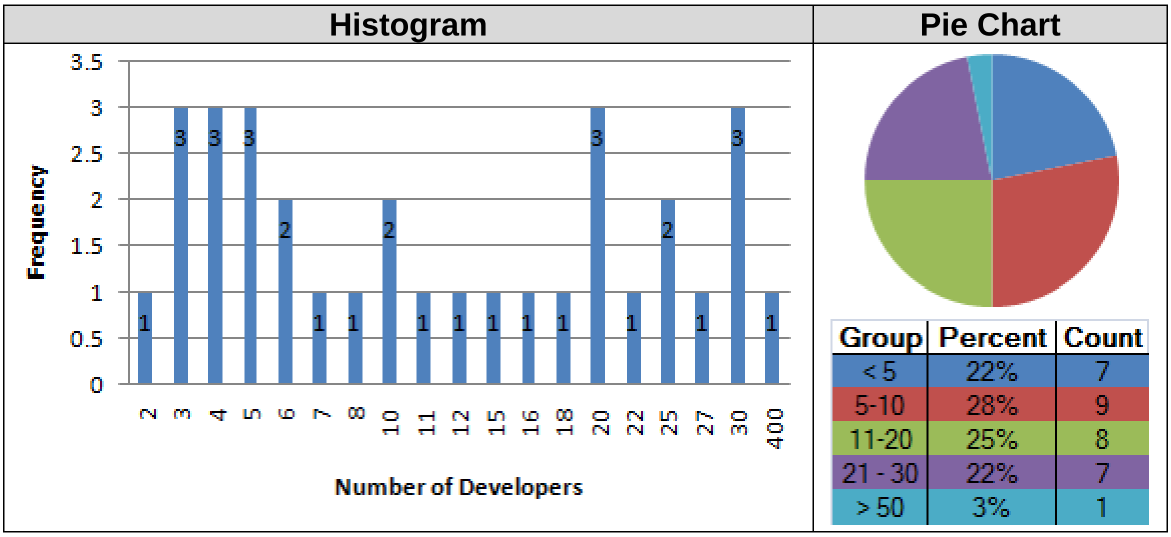
The number of software development teams in each company varies. All except for two companies have more than two software development teams. The most frequent team size in a company is three software development teams with the highest of 30 software development teams in two companies as shown in the histogram in Figure 5.2 below. This wide range of team sizes supports the unit of analysis at the team level.

Figure 5.2: Number of Software Development Teams



The number of developers in each team varies between two and 30 developers with the exception of one team of 400 developers as shown in Figure 5.3 below. The data implies that half of the teams are suitable for the Agile development approach based on team size as 50% of the teams have less than ten developers.

Figure 5.3: Number of Developers

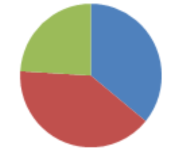


Group	Percent	Count
< 5	22%	7
5-10	28%	9
11-20	25%	8
21 - 30	22%	7
> 50	3%	1

The average time which the teams that responded to the survey spend between new product development, enhancements to existing products and maintenance of existing products is 36%, 40% and 24% respectively as shown in Figure 5.4 below. All teams work on all three types of product development except for three teams which do not do new product development. As most teams engage in a similar spread of product development, no influences could be found between the type of product development and the methodology selected. Therefore, the type of product development was not analysed further and is not regarded as a factor influencing the selection of software development methodologies.

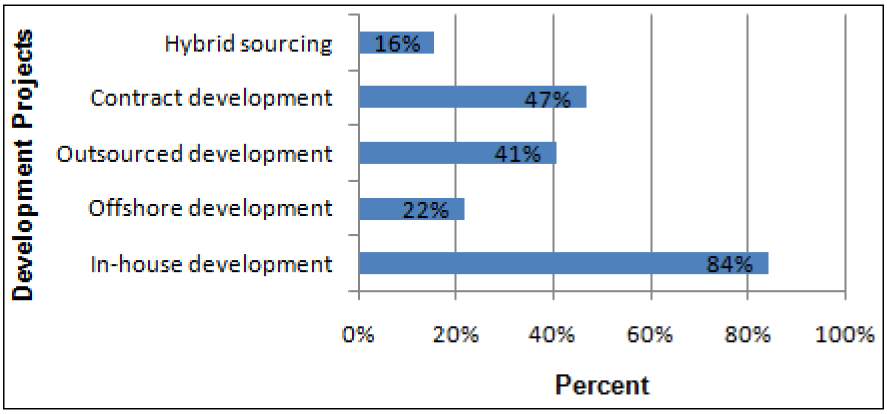
Figure 5.4: Types of Product Development

Product Type	Percent
New products.	36%
Enhancements to existing products (new features).	40%
Maintenance on existing products (bug fixes).	24%



All teams are involved in multiple types of software development projects with the most frequent being 84% of teams which do in-house development as depicted in Figure 5.5 below. The other four types of development projects feature less prominently, therefore, it is viable to investigate if these types of projects influence methodology selection. Further analysis on the type of projects revealed no direct link towards the selection of a specific software development methodology. Therefore, the type of projects was not analysed further and is not regarded as a factor influencing the selection of software development methodologies.

Figure 5.5: Types of Software Development Projects



5.4. Current Trends in Adopting Software Development Methodologies

This section aims to answer all three research questions with regards to the current trends in adopting software development methodologies.

5.4.1. Research Question 1

*What are the **current** and future trends in adopting software development methodologies in the South African software development industry?*

Only the first part of research question one is answered in this section with the second part of the question discussed under section 5.5 below.

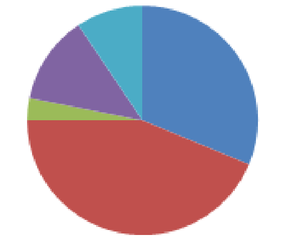
5.4.1.1. Current Dominant Methodology

The most dominant methodology adopted in the South African software development industry is the Agile methodology which is utilised by 44% of respondents as depicted in Figure 5.6 below. The Waterfall approach is the next most frequently used methodology with 31% of respondents using this methodology.

Only one team has adopted CMMI as a dominant methodology while four teams do not follow any software development methodology. The three respondents who selected the other option specified the methodology used as Application Implementation Methodology, Rapid Application Development and Service Orientated Architecture Methodology.

Figure 5.6: Current Dominant Methodology Used

Dominant Methodology	Percent	Count
Waterfall	31%	10
Agile Methods	44%	14
CMMI	3%	1
None	13%	4
Other (please specify)	9%	3

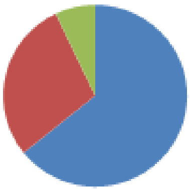


Since Agile methods are the most dominant software development methodology used, it is important to note that the most dominant Agile method is the Scrum methodology, which 11 of the 14 teams following the Agile method have adopted. The prescriptive characteristics for choosing the Scrum method listed in Table 2.1 are, therefore, more relevant factors for methodology selection. These factors will be analysed further in section 5.4.3 below.

Of the 14 teams that have adopted the Agile method, 64% have done so for less than two years, while 29% of teams have been using the Agile method between two and five years as shown in Figure 5.7 below. Only one team has adopted the Agile method for more than five years.

Figure 5.7: Period Agile Method Used

Period	Count	Percent
< 2 years	9	64%
2-5 years	4	29%
> 5 years	1	7%
Total	**14**	**100%**



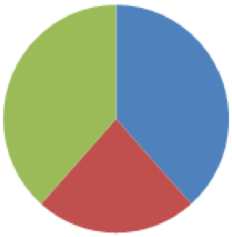
The decision to adopt the Agile method in the 14 teams is a 50-50 split between the company and the team manager. Furthermore, all eight responses where the methodology was selected by the company revealed that the Agile method is the dominant methodology used. This shows a tendency towards the adoption of the Agile method at a company level.

5.4.1.2. Current Secondary Methodology

The majority of teams adopt more than one methodology as shown in Figure 5.8 below with 61% of teams adopting a secondary methodology. The most frequent secondary methodology adopted is the Agile method with the Waterfall method being the other methodology used. It is important to note that all teams that use the Waterfall method as the secondary methodology also adopt the Agile method as the dominant methodology and vice versa. The one team using CMMI as the dominant methodology has adopted the Agile method as a secondary methodology.

Figure 5.8: Current Secondary Methodology Used

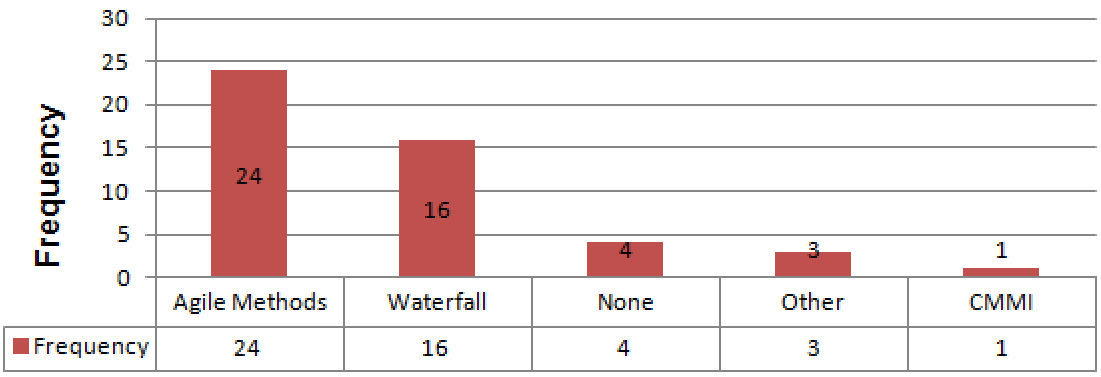
Answer Options	Percent	Count
None	38.5%	10
Waterfall	23%	6
Agile Methods	38.5%	10
CMMI	0.0%	0
Other (please specify)	0.0%	0



5.4.1.3. Summary of Current Trends

The Agile method is the most frequently used software development methodology with 24 of the 32 teams using this methodology as either their dominant or secondary methodology as shown in Figure 5.9 below. The Waterfall method is the next frequently used methodology with 16 of the 32 teams using this as either their dominant or secondary methodology.

Figure 5.9: Total Methodology Usage



5.4.2. Research Question 2

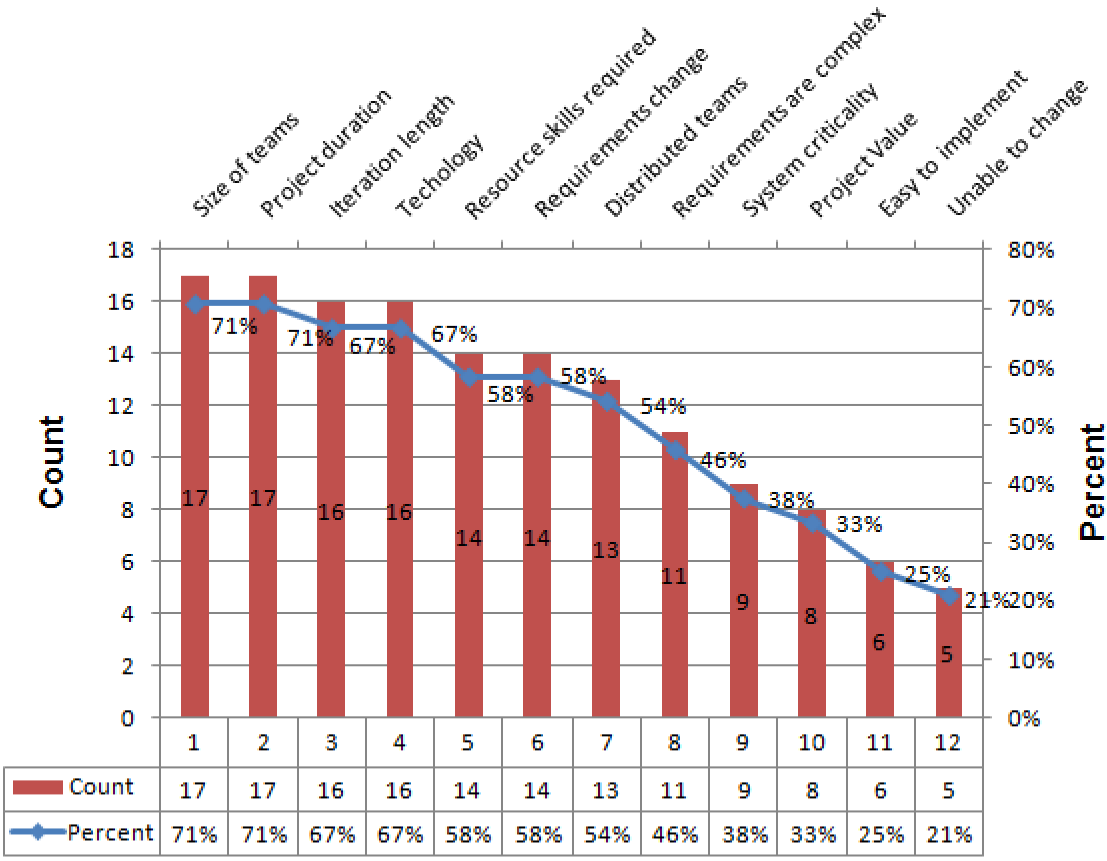
What are the factors influencing the selection of software development methodologies in the South African software development industry?

The findings in section 5.4.1 above provide evidence towards the use of two software development methodologies with the Agile method more widely used than the Waterfall method. This section explores the factors influencing the selection of these two methodologies. CMMI has been excluded from this analysis as only one team has adopted this methodology in the sample.

5.4.2.1. Factors Influencing the Selection of Agile Methods

The most frequent factors influencing the selection of Agile methodologies are discussed in this section. A total of 24 teams use the Agile method as either their dominant or secondary methodology. A summary of the combined factors for selecting the Agile method as a dominant or secondary methodology is shown in Figure 5.10 below. Factors which were selected by more than half of the respondents using Agile methods are more significant and are regarded as the factors influencing the selection of the Agile methods. Therefore, factors selected by less than 12 respondents were excluded. The percentages shown in the table are a percentage of the 24 teams using the Agile method and not of the 32 teams that responded to the survey.

Figure 5.10: Factors Influencing the Selection of Agile Methods



The data shows that the two most important factors for selecting the Agile method are the size of the development team and the duration of projects which the team is involved in. Nine of the 17 respondents that selected team size as a factor have teams of between one and five members.

Of the 17 respondents that selected project duration as a factor, 14 of them work on projects where the duration is less than six months. The remaining three respondents use the Agile method for projects where the duration is greater than six months.

The data revealed that 12 of the 16 respondents who selected iteration length as a factor have iterations of between two and four weeks. New technology is being used in the teams of ten of the 16 respondents that selected technology as a factor for following the Agile method. Therefore, iteration length and new technology are both distinguishing factors influencing the selection of Agile methods.

There is a 50-50 split between medium skills and high skills required as a factor for selecting the Agile method. None of the respondents listed low skills as a factor for selecting the Agile method. The last two factors considered as being significant factors influencing the selection of the Agile method, are, changing requirements and support for distributed teams which was selected by 14 and 13 respondents respectively.

The remaining five factors were selected by less than half of the respondents using the Agile method. These factors include the complexity of requirements, criticality of systems and ease of implementation selected by 11, nine and six respondents respectively. The other factor is the value of projects selected by eight respondents who entered varying project values starting from less than a hundred thousand rand to over one million rand. The last factor selected by five respondents is the inability to change with the lack of time, resources and budget listed as the three reasons.

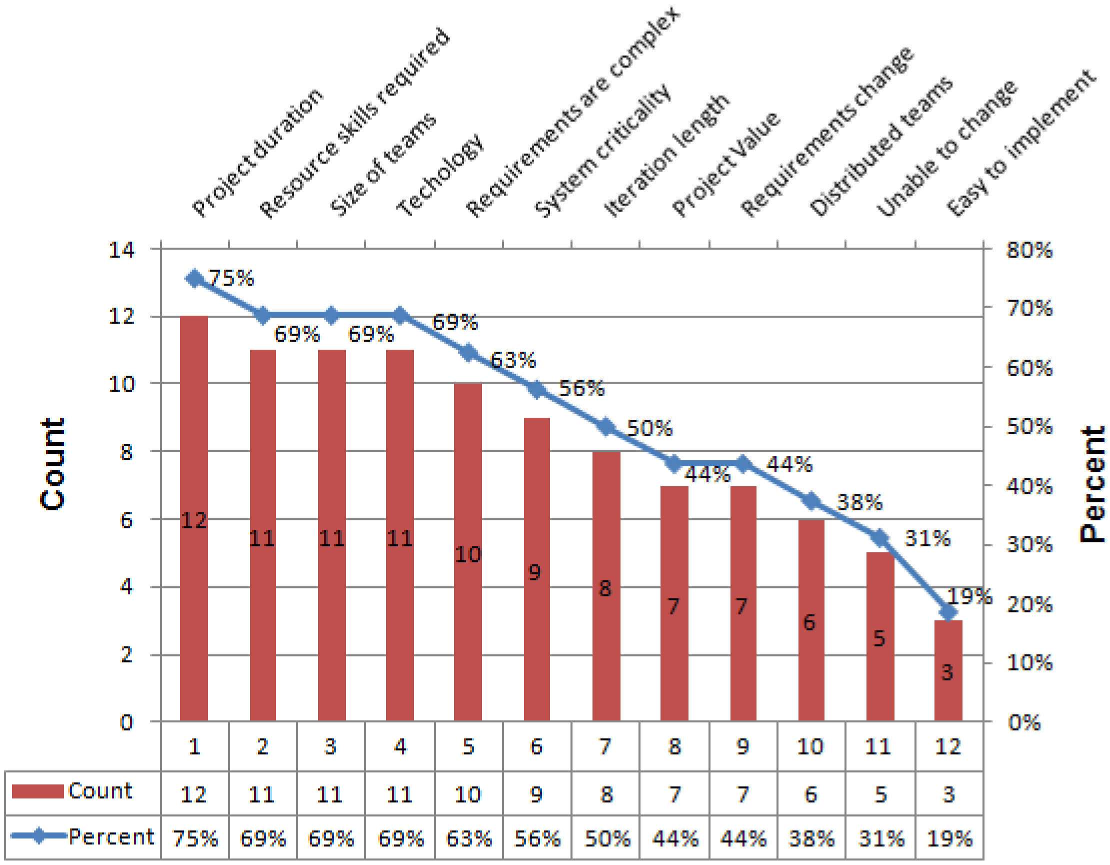
The Other option was selected by two respondents who stated the reason for adopting the Agile method is to improve communication as well as the quality of work. These factors are outside the scope of this research as they were not included as a response option in the survey, however, they are considered as ideas for future research.

5.4.2.2. Factors Influencing the Selection of the Waterfall Method

The most frequent factors influencing the selection of the Waterfall method are discussed in this section. A total of 16 of the 32 teams use the Waterfall method as either their dominant or secondary methodology. A summary of the combined factors for selecting the Waterfall method as a dominant or secondary methodology is shown in Figure 5.11 below.

Factors which were selected by more than half of the respondents using the Waterfall method are more significant and are regarded as the factors influencing the selection of the Waterfall method. Therefore, factors selected by less than eight respondents were excluded. The percentages shown in the diagram are a percentage of the 16 teams using the Waterfall method and not of the 32 teams that responded to the survey.

Figure 5.11: Factors Influencing the Selection of the Waterfall Method



	1	2	3	4	5	6	7	8	9	10	11	12
🟥 Count	12	11	11	11	10	9	8	7	7	6	5	3
◆ Percent	75%	69%	69%	69%	63%	56%	50%	44%	44%	38%	31%	19%

The data shows that the most important factor for selecting the Waterfall method is the duration of projects which the team is involved in. The eight respondents from the 12 that selected project duration as a factor have projects running for more than three months.

The next three important factors for selecting the Waterfall method are resource skills, size of the teams and the technology used by the team. The 11 respondents that selected resource skills indicated that senior and medium resources are required, which is similar to the skills selected for the Agile method.

The Waterfall method is used by teams with a varying number of team members from less than five members to greater than 15 members. In comparison to the Agile method which had a greater tendency towards small teams of between one and five members, the size of the team cannot be regarded as a distinguishing factor influencing the selection of the Waterfall method.

There is a similar frequency of responses in terms of the technology used with six respondents selecting old technology as a factor, while five respondents selected new technology as a factor for selecting the Waterfall method. In comparison to the data for Agile methods, this implies that the type of technology being used is not a distinguishing factor in selecting the Waterfall method.

The complexity of requirements was selected as a factor influencing the use the Waterfall method by 10 respondents. Similarly, nine respondents selected the criticality of systems as a factor for selecting the Waterfall method.

The next frequent factor for selecting the Waterfall method is the iteration length. However, there are varying responses with regards to the length of each iteration, ranging from less than two weeks to greater than three months. The iteration length is, therefore, not a distinguishing factor influencing the selection of the Waterfall method.

The remaining five factors were selected by less than half of the respondents using the Waterfall method and are not regarded as factors influencing the selection of the Waterfall method. These factors include changing requirements, support for distributed teams, inability to change and ease of implementation which was selected by seven, six, five and three respondents respectively. The three respondents that selected inability to change from the Waterfall method listed the lack of time and budget as the reason. The other factor is the value of projects which was selected by seven respondents with varying project values specified.

5.4.3. Research Question 3

Are the software development methodologies adopted by companies in the South African software development industry effective in delivering successful projects locally and internationally?

5.4.3.1. Measuring Project Success

Only three respondents do not measure the success of projects in their team. The remaining respondents measure the success of projects in various ways. The summary rank ordering was used to calculate the ranking preference for the three measurements for project success. The rankings depicted in Table 5.1 below show that the respondents rank Time as the most important measurement for project success, followed by Scope and then Budget.

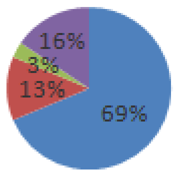
Table 5.1: Ranking Preference for Measuring Project Success

Rank	Project Measurement	Total Ranking Score
1	Time	42
2	Scope	52
3	Budget	62

5.4.3.2. Project Success

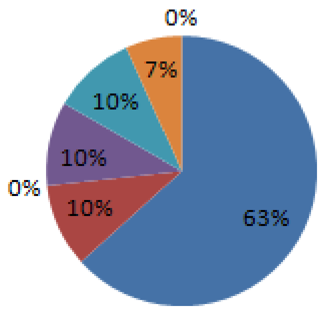
The majority of respondents (69%) believe that their dominant methodology effectively delivers successful projects, while 13% of respondents do not believe their dominant methodology delivers successful projects as shown in Figure 5.12 below.

Figure 5.12: Does Dominant Methodology Deliver Successful Projects

Answer Options	Percent	Count	
Yes	69%	22	
No	13%	4	
No Methodology Used	3%	1	
Not Measured	16%	5	

The data shows that 63% of teams do not tender or quote for new software development projects as shown in Figure 5.13 below. Of the teams that tender for new work, 10% of teams do not measure the percentage of tenders won, while 20% win more than 50% of the tenders that they participate in. In addition, 7% of teams have less than 50% success rate in winning tenders.

Figure 5.13: Tenders Won

Answer Options	Percent	Count	
Do not tender (team does not tender or quote for projects)	63%	19	
Not measured (percentage of tenders or quotes won are not measured)	10%	3	
100% won	0%	0	
75-99% won	10%	3	
50-74% won	10%	3	
25-49% won	7%	2	
<25% won	0%	0	

The data shows that 17 of the 24 respondents using the Agile method selected Time as the most important measurement of project success, while 20 of the respondents believe that the factors for selecting the Agile method are effective in delivering projects successfully. Similarly, 11 of the 16 respondents using the Waterfall method selected Time as the most important measurement of project success, while 12 of the respondents believe that the factors for selecting the Waterfall method are effective in delivering projects successfully.

Of the 22 teams that believe their dominant methodology delivers projects successfully as shown in Figure 5.12 above, 14 of these teams use Time as the most important measurement of project success. Of these 14 teams, nine of the teams adopt the Agile method, while five of the teams adopt the Waterfall method. This implies that the Agile method is the preferred methodology used to deliver projects on time.

5.5. Future Trends in Adopting Software Development Methodologies

This section aims to answer the first two research questions with regards to the future trends in adopting software development methodologies.

5.5.1. Research Question 1

*What are the current and **future** trends in adopting software development methodologies in the South African software development industry?*

The second part of research question one is answered in this section with the first part of the question discussed under section 5.4.1 above.

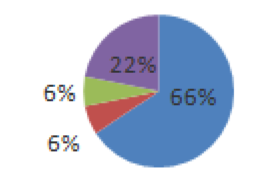
5.5.1.1. Future Methodology

In general, the majority of respondents (66%) intend on using their current dominant methodology in the future, while 6% of teams plan to use their current secondary methodology as their dominant methodology in the future as shown in Figure 5.14, and discussed below. One team using the Agile method currently, does not plan on using any methodology in the future.

The two respondents who selected the current secondary methodology and one respondent who selected a different methodology have all listed the Agile method as the choice of methodology in the future. The other respondent that selected a different methodology plans on using a combination of the Waterfall method, Agile method and CMMI.

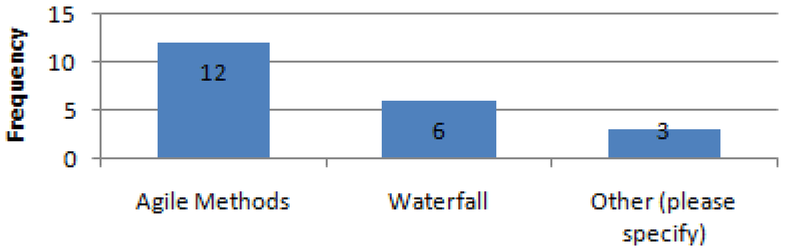
Figure 5.14: Future Methodology Usage

Answer Options	Percent	Count
Current dominant methodology	66%	21
Current secondary methodology	6%	2
Different methodology	6%	2
None	22%	7



The current dominant methodologies, which 66% of respondents plan to continue using in the future, are shown in Figure 5.15 below. This shows a trend towards greater adoption of the Agile method in the future, the Scrum method in particular, which was highlighted in section 5.4.1 above. The Waterfall method is also a common choice, with six respondents planning on using this method in the future. However, there is very little intention of adopting CMMI with only one respondent planning on using CMMI in the future.

Figure 5.15: Current Dominant Methodology Usage in the Future



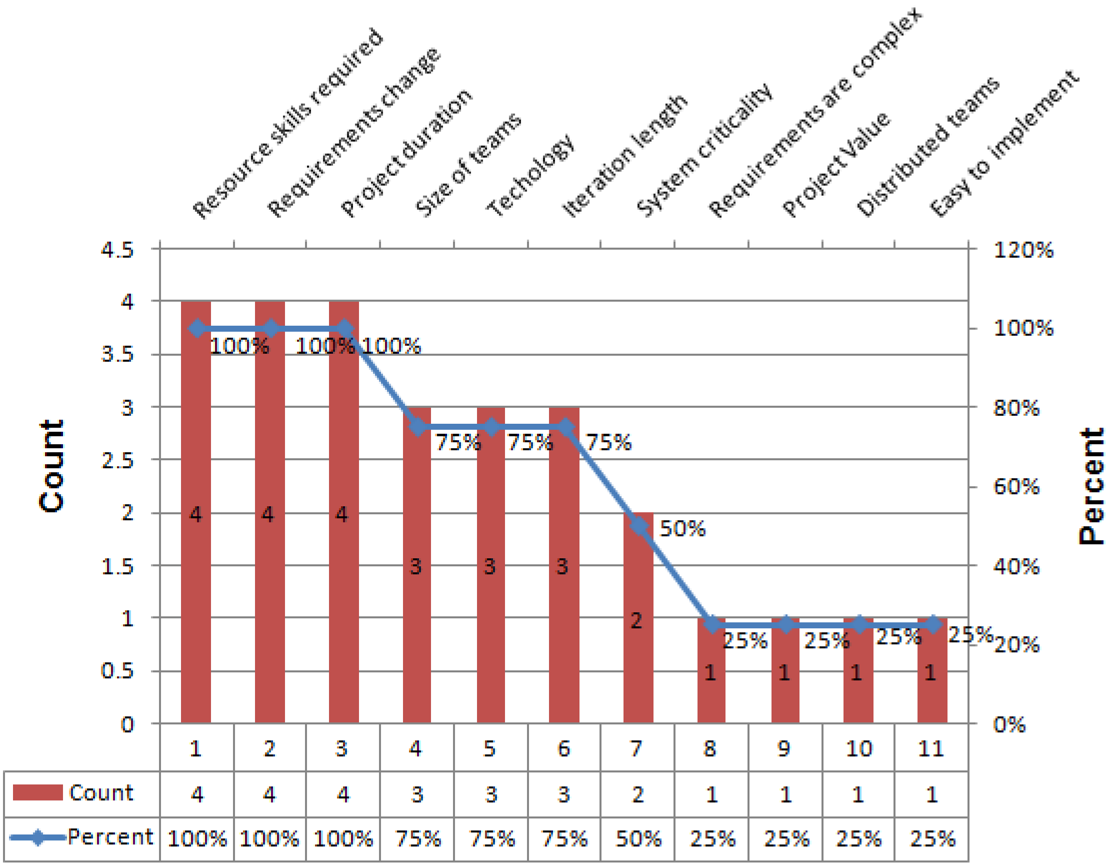
5.5.2. Research Question 2

What are the factors influencing the selection of software development methodologies in the South African software development industry?

The factors influencing the selection of the Waterfall method and Agile methods as either a dominant or secondary methodology were described above. The factors influencing respondents to select a different methodology to what they are currently using, are described below.

Three respondents not using any software development methodology and one respondent using the Waterfall method currently, have all selected the Agile method as the different methodology they would like to adopt in the future. The factors for wanting to adopt this different methodology in the future is summarised in Figure 5.16 below.

Figure 5.16: Factors Influencing the Selection of Agile Methods in the Future



	1	2	3	4	5	6	7	8	9	10	11
Count	4	4	4	3	3	3	2	1	1	1	1
Percent	100%	100%	100%	75%	75%	75%	50%	25%	25%	25%	25%

All four respondents selected resource skill as a factor for selecting the Agile method. However, two respondents selected medium skills, while one respondent selected high skills and one low skills. All four respondents selected changing requirements as a factor for wanting to adopt the Agile method in the future.

The four respondents also selected project duration as a factor for wanting to adopt the Agile method. The duration of projects vary with two respondents selecting project durations between one and three months, one respondent selecting project durations between three and six months and the other respondent selecting project durations greater than six months.

Three respondents selected team size, technology and iteration length as factors for wanting to adopt the Agile method in the future. In all three cases, two of the three responses supported the previous findings in section 5.4.2.1. Therefore, these are regarded as factors influencing the use of the Agile method in the future.

The remaining four factors were only selected by one respondent, and are therefore excluded, as they cannot be compared to other responses in this category. The previous analysis in section 5.4.2.1 regarding these factors is, therefore, applicable.

5.6. Conclusion

This chapter presented the results gathered from the survey. The data shows a strong adoption of the Agile method as the primary and secondary methodology. The Agile method is also the dominant methodology that teams plan on using in the future. It is common for companies to adopt more than one methodology and the Waterfall method is the next methodology most widely used. There is little adoption towards CMMI.

The most dominant factors influencing the selection of Agile methods is small team sizes of between one and five members, the complexity of changing requirements and the duration of projects of less than three months.

Most companies measure the success of projects, with the most important measurement being Time. The majority of respondents indicated that the Agile method (nine teams) compared to the Waterfall method (five teams) is effective in delivering projects on time.

6. CHAPTER 6: DISCUSSION OF RESULTS

6.1. Introduction

This chapter aims to meet the objectives of the research by answering all three research questions defined in chapter three. Each research question is presented under a separate heading and discussed in terms of the data, literature reviewed and the research topic.

6.2. Research Question 1

What are the current and future trends in adopting software development methodologies in the South African software development industry?

The literature revealed three categories which the various software development methodologies can be grouped into (Jiang & Eberlein, 2008; Paul, 2008; Pikkarainen, 2008; Vinekar, Slinkman & Nerur, 2006). Further comparison of past studies narrowed the wide range of methodologies to a dominant methodology in use from each of the three categories. The survey results support the literature and selection of these three methodologies as 89% of respondents using a methodology have either adopted the Waterfall method, Agile method or CMMI within the South African context.

The data revealed that only four respondents are not using any software development methodology currently. This shows an increase in the adoption of software development methodologies in the last twelve years since the last study by Fitzgerald in 1998 which found that 40% of respondents were not using a formal software development methodology. Fitzgerald (1998) found that 79% of those respondents not using a methodology did not intend to adopt one in the future. The data contradicts this finding by Fitzgerald (1998) by providing strong evidence in favour of using a formal software development methodology. This is supported by further analysis of the data which found that three of the four respondents not using a methodology currently plan on adopting the Agile method in the future.

The data provides strong evidence in the use of more than one methodology. The data, therefore, supports the studies by Vinekar et al. (2006) and Khalifa and Verner (2000) which found that Agile methods and traditional methods can coexist, as organisations prefer to sustain both forms of development. The statement by Paul (2008) that the Waterfall method is still widely in use is therefore applicable to the South African context.

The data shows little adoption of CMMI, contrary to the view of the JCSE. However, the data also revealed that 63% of respondents do not tender or quote for new projects, the holding company of 88% of respondents are South African and only 22% of respondents engage in offshore software development projects. These three factors are listed by the JCSE (2010) as the main reason for the adoption of CMMI by US and European companies that award offshore tenders. The data therefore suggests that the majority of South African companies engage in local software development projects which do not necessitate the need for adopting CMMI.

In summary, the current trend shows a greater preference towards the use of Agile methods in the South African context, however, the need for traditional methods still exists with a large number of teams preferring to utilise the Waterfall method as a dominant or secondary methodology. The future trend shows an increased adoption of Agile methods as the dominant methodology however teams will still prefer to use both forms of development. The data also provides strong evidence towards the use of the Scrum method in particular as the preferred Agile methodology.

Coleman and O'Connor (2008) argue that when considering changes in software development methodologies, previous experience can influence the selection of a methodology. The data supports this argument and revealed that in all instances where the decision to adopt a methodology is a company decision, the Agile method is followed. This shows the tendency to adopt the Agile method at the company level within the South African context.

6.3. Research Question 2

What are the factors influencing the selection of software development methodologies in the South African software development industry?

The data provides evidence that all factors influence the selection of software development methodologies as each factor was selected by at least one respondent for both the Waterfall and Agile methods. However, the distinguishing factors influencing the selection of Agile methods as opposed to the Waterfall method in the South African context was analysed and the results are discussed below.

6.3.1. Distinguishing Factors Influencing Methodology Selection

The data provides strong evidence that the size of teams is a major factor influencing the selection of Agile methods. The most common team size is between one and five team members. This supports team size as one of the four prescriptive characteristics listed by Cohen et al. (2004) for adopting the

Agile method. Team size was not found to be a distinguishing factor influencing the selection of the Waterfall method.

Batra (2009) argues that companies tend to offshore large projects at a lower cost which creates opportunities for software development companies to expand their markets. The data shows that the majority of companies engage in local software development projects only. Although very few companies engage in large projects that are greater than six months, the data provided strong evidence towards the adoption of Agile methods when the project duration is less than three months and the Waterfall method when the project duration is greater than three months. The duration of projects is therefore a distinguishing factor influencing the selection of software development methodologies.

The data provides strong evidence towards the adoption of Agile methods when the length of each iteration is between two and four weeks. This supports iteration length as one of the four prescriptive characteristics listed by Cohen et al. (2004) for adopting the Agile method. The prescriptive characteristics further indicate that the Scrum method is the most suitable method when the iteration length is between two and four weeks. The data also supports this characteristic listed by Cohen et al. (2004). Iteration length was not regarded as a distinguishing factor influencing the selection of the Waterfall method.

The data shows a greater frequency for adopting the Agile method when new technology is used. Technology is not a distinguishing factor for selecting the Waterfall method. The data, therefore, supports the findings by Nerur et al. (2005) with regards to Agile methods being more ideal for projects involving new technology.

6.3.2. Factors Not Influencing Methodology Selection

The data shows that the teams engage in various types of projects and product development, however, there was no evidence which linked a type of project or product development to the adoption of a specific methodology.

Nerur et al. (2005) supports the skills of resources as a factor influencing the selection of Agile methods. The data provided strong evidence for medium and high skills required for both the Waterfall and Agile methods. Therefore, resource skills are not a distinguishing factor that influences the selection of the Agile method over the Waterfall method.

Williams (2010) argues that the Agile method welcomes changing requirements, even late in development. Nerur et al. (2005) also argues that Agile methods are used increasingly to address changing requirements. While the data does support both Williams (2010) and Nerur et al. (2005) there is also strong evidence that suggests that changing requirements is also a factor influencing the selection of the Waterfall method in the South African context. This can be interpreted as teams either using the Waterfall method to stop requirements from changing or allowing requirements to change earlier rather than later on in development.

The results provide evidence that both the Agile and Waterfall method are used with distributed teams. A higher frequency in response suggests that there is a greater tendency to adopt the Agile method when supporting distributed teams. The study by Cohen et al. (2004) defines distributed support as one of the four factors governing which Agile method to select. The data, therefore, supports this statement that distributed teams influences which Agile method to use rather than influencing the selection of the Agile methodology over other software development methodologies. Distributed teams are therefore not regarded as a distinguishing factor influencing methodology selection.

The literature found that the complexity of products are increasing (SEI, 2006) and traditional software development methodologies cannot be used to build complex systems in a single pass without revisiting requirements (Szalvay, 2008). Lee and Xia (2010) argue that Agile methodologies are known for managing complex requirements. The data contradicts these studies and provides evidence that the Waterfall method and Agile methods are both used to handle complex requirements in the South African context.

Cohen et al. (2004) lists system criticality as one of the prescriptive characteristics governing the choice of Agile method to use. The data suggests that both the Agile and Waterfall methods are used to deliver critical systems. This implies that system criticality influences the choice of Agile method rather than influencing the selection of the Agile methodology over other software development methodologies. System criticality is therefore not a distinguishing factor influencing the selection of a specific methodology.

Despite budget being an important measurement of project success as defined by the literature (The Standish Group, 1995), the data suggests that the value of projects is not a distinguishing factor influencing the selection of a particular methodology.

Paul (2008) argues that traditional methodologies are still widely used because they are easy to adopt. The data provides evidence that both the Waterfall and Agile methods are used because they are easy to implement. This may be one of the reasons why CMMI is not widely adopted in South Africa following the argument by Niazi and Babar (2009) that CMMI is not easy to implement, especially in small and medium size companies.

6.4. Research Question 3

Are the software development methodologies adopted by companies in the South African software development industry effective in delivering successful projects locally and internationally?

The data provides strong evidence that the most important measurement of project success within the South African context is delivering the project on time. This contradicts the study by Agarwal and Rathod (2005) which found that scope is the most important factor for project success. Therefore, in the South African context, both the Waterfall and Agile methods are effective in delivering projects successfully in relation to time, which is the most important factor for project success.

6.5. Conclusion

This chapter discussed the results in relation to the research questions and literature. The results revealed the adoption of the Agile methodology as the current dominant methodology and the future methodology. This study supports the literature and found that it is common for companies to adopt more than one methodology and the Waterfall method is the secondary methodology most widely used.

The data suggests that all factors identified in chapter two influence methodology selection since all factors were selected at least once for both the Waterfall and Agile methods. However, the distinguishing factors influencing the selection of Agile methods over the Waterfall method in the South African context is team size, project duration, iteration length and technology used, as summarised in Table 6.1 below.

Table 6.1: Distinguishing Factors Influencing Methodology Selection

Factors	Agile Methods	Waterfall Method
Team Size	1-5 members	Not applicable
Project Duration	< 3 months	< 6 months
Iteration Length	2-4 weeks	Not applicable
Technology	New technology	Not applicable

Time is the most important measurement of project success in the South African context which differs from the literature reviewed. The majority of respondents indicated that their current dominant methodology, the Agile method in particular, is effective in delivering projects successfully.

7. CHAPTER 7: CONCLUSION

7.1. Introduction

This concluding chapter of the report summarises the main findings of the research and highlights the contribution of this study to both business and academia. The limitations of this study are discussed as well as recommendations and ideas for future research in this area.

7.2. Main Findings

This research aimed to meet the following objectives:

- **Objective 1**: to describe the state of the software development industry in South Africa with regards to adoption of software development methodologies.
- **Objective 2**: to determine which factors influence the selection of software development methodologies in the software development industry in South Africa.
- **Objective 3**: to determine if the software development methodologies adopted in the software development industry in South Africa are effective in delivering successful projects both locally and internationally.

The study found that the Agile method is the most dominant methodology adopted in the software development industry in South Africa. The Agile method is also the methodology of choice that teams plan on using in the future.

It is common practice within the South African context for companies to adopt more than one software development methodology and the Waterfall method is the next methodology most widely used. There is little adoption towards CMMI.

The data suggests that all factors identified in chapter two influences the selection of software development methodologies. However, the distinguishing factors influencing the selection of Agile methods over the Waterfall method in the South African context is team size of between one and five members, project duration of less than three months, iteration length of between two and four weeks and new technology, as summarised in Table 7.1 below.

Table 7.1: Summary of Key Findings

Factors	Agile Methods	Waterfall Method
Team Size	1-5 members	Not applicable
Project Duration	< 3 months	< 6 months
Iteration Length	2-4 weeks	Not applicable
Technology	New technology	Not applicable

The success of projects is measured by most companies. The most important measurement of project success within the South African context is the delivery of projects on time. The majority of respondents believe that their current dominant methodology, the Agile method in particular, is effective in delivering projects on time. However, these companies mostly engage in local software development projects only.

7.2.1. Contribution of This Study

This was the first research study that combined the most widely used software development methodologies and evaluated their adoption within the software development industry in a single country. This study supports the relevance of the current literature on the Waterfall method, Agile methods and CMMI, however, CMMI is less relevant to the South African context. This study confirms the demise of all other software development methodologies in practice.

The study demonstrates that there isn't a silver bullet when it comes to selecting a software development methodology and provides academia with the challenge of developing a methodology that is inclusive of the benefits and characteristics of all methodologies currently in use. An alternative is to develop a framework with the selection criteria for choosing a methodology. This will assist managers to follow a structured process or checklist to determine the conditions under which to select a specific methodology.

7.2.2. Implication for Software Development Managers

These findings show that the most important measurement of project success within the South African context is delivering projects on time. The majority of companies are using Agile methods to deliver projects successfully, however, they also follow the Waterfall method as a second methodology. It is therefore important for Software Development Managers to understand the critical success factors for each project and then determine which of the two methodologies will be suitable for delivering the project according to the specific success factors for that project.

7.3. Recommendations for Future Research

This study used descriptive statistics to determine the level of adoption and effectiveness of software development methodologies in the software development industry in South Africa. The study found that the Agile and Waterfall methods are the two methodologies most adopted and most effective. Further research can now be conducted to statistically test these findings using the results as the hypothesis. In addition, a framework for selecting a methodology can be developed and tested as discussed in section 7.2.1 above.

This study confirmed that all factors influence the selection of methodologies as each factor was selected at least once. Future research can use a Likert scale to determine how strongly each factor influences the selection of software methodologies. The other option is to use ranking ordering to determine which factors are more important in relation to each other.

Two respondents provided two additional factors for adopting the Agile method. These factors are, to improve communication and the quality of work. Future research should determine the significance of these factors in influencing the selection of the Agile method as well as the other methodologies.

7.4. Limitations

The link to the online survey was sent to 46 team managers from 22 companies with a total of 32 responses received. In order to achieve a high response rate, the survey was completely anonymous, therefore, the number of companies that did respond could not be determined.

Although the sample was representative of the population for the purposes of descriptive analysis and included both large and small software development companies in South Africa, a larger sample would have been more beneficial. This will be required to statistically test the findings in the future research.

8. REFERENCE LIST

Agarwal, N. & Rathod, U. (2005). Defining 'success' for software projects: an exploratory revelation. *International Journal of Project Management*, 24(2006), 358-370. doi:10.1016/j.ijproman.2005.11.009

Awad, M. A. (2005). A comparison between agile and traditional software development methodologies. Retrieved from http://pds10.egloos.com/pds/200808/13/85/A_comparision_between_Agile_and_Traditional_SW_development_methodologies.pdf

Batra, D. (2009). Modified agile practices for outsourced software projects. Communication of the ACM, 52(9), 143-148. doi: 10.1145/1562164.1562200

Cohen, D., Lindvall, M. & Costa, P. (2004). An introduction to agile methods. *Advances in Computers*, 62. dio:10.1016/S0065-2458(03)62001-2

Coleman, G. & O'Connor, R. V. (2008). An investigation into software development process formation in software start-ups. *Journal of Enterprise Information Management*, 21(6), 633-648. doi:10.1108/17410390810911221

Cusick, J. J., Prasad, A., & Tepfenhart, W. M. (2008). Global software development: origins, practices, and directions. *Advances in Computers*, 74, 201-269. doi:10.1016/S0065-2458(08)00606-2

Eveleens, J. L. & Verhoef, C. (2008). *The rise and fall of the chaos report figures.* Amersterdam: VU University Amsterdam.

Fitzgerald, B. (1998). An empirical investigation into the adoption of system development methodologies. Information and Management, 34, 317-328. Retrieved from: http://0-www.sciencedirect.com.innopac.up.ac.za/science?_ob=ArticleURL&_udi=B6VD0-3V92MM3-2&_user=59388&_coverDate=12%2F21%2F1998&_rdoc=1&_fmt=high&_orig=search&_sort=d&_docanchor=&view=c&_searchStrId=1403357276&_rerunOrigin=scholar.google&_acct=C000005298&_version=1&_urlVersion=0&_userid=59388&md5=9ec9bb47d5bda60a970e6e4405dc9f69

Glazer, H., Dalton, J., David Anderson, David J. Anderson & Associates Inc, Konrad, M., Shrum, S. (2008). *CMMI or Agile: Why Not Embrace Both*. Hanscom: Software Engineering Institute. Available from: http://www.sei.cmu.edu/reports/08tn003.pdf (accessed 01/03/2010)

Glass, R. L. (2006) The Standish report: does it really describe a software crisis? *Communications of the ACM*, 49(8), 15-16. Retrieved from: http://0-delivery.acm.org.innopac.up.ac.za/10.1145/1150000/1145301/p15-glass.pdf?key1=1145301&key2=7488739721&coll=GUIDE&dl=GUIDE&CFID=97332818&CFTOKEN=75012371

Griffin, A. S. & Brandyberry, A. A. (2008). *System Development Methodology Usage in Industry: A Review and Analysis*. Available from: http://proc.conisar.org/2008/3522/CONISAR.2008.Griffin.pdf (accessed 01/03/2010)

Jiang, L. & Eberlein, A. (2008). *Towards a Framework for Understanding the Relationships between Classical Software Engineering and Agile Methodologies*. New York: ACM. Available from: http://0-portal.acm.org.innopac.up.ac.za/citation.cfm?id=1370146 (accessed 01/03/2010)

Johannesburg Centre for Software Engineering (2010). *Why CMMI*. Available from http://www.jcse.org.za/cmmi.php?sectionid=47 (accessed 01/03/2010)

Khalifa, M., & Verner, J. M. (2000). Drivers for software development method usage. *IEEE Transactions on Engineering Management*, 47(3), 360-369. Retrieved from http://0-teaching.fec.anu.edu.au.innopac.up.ac.za/infs8005/R106.pdf

Lee, G. & Xia, W. (2010). Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility. *MIS Quarterly*, 34(1), 87-114. Retrieved from http://aisel.aisnet.org/misq/vol34/iss1/7/

Liu, J. Y., Chen, V. J., Chan, C., & Lie, T. (2008). The impact of software process standardization on software flexibility and project management performance: control theory perspective. *Information and Software Technology*, 50(2008), 889–896. doi:10.1016/j.infsof.2008.01.002

Nerur, S., Mahapatra, R. & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, 48(5), 73-78. Retrieved from http://0-www.umsl.edu.innopac.up.ac.za/~sauter/analysis/challenges_of_migrating_to_agile_methodologies.pdf

Niazi, M. & Babar, M. A. (2008). Identifying high perceived value practices of CMMI level 2: An empirical study. *Information and Software Technology*, 51(2009), 1231-1243. doi:10.1016/j.infsof.2009.03.001

Paul, J. (2008). *Quantitative Approach for Lightweight Agile Process Assessment* (Master's thesis, University of Turku). Available from: http://www.johanpaul.com/thesis_johanpaul.pdf (accessed 01/03/2010)

Pikkarainen, M. (2008). *Towards a Framework for Improving Software Development Process Mediated with CMMI Goals and Agile Practices*. (Academic dissertation, University of Oulu). Available from: http://www.vtt.fi/inf/pdf/publications/2008/P695.pdf (accessed 01/03/2010)

Ravichandran, R. & Ahmed, N. U. (1993). Offshore systems development. *Information and Management*, 24, 33-40. doi:10.1016/0378-7206(93)90045-U

Rubinstein, D. (2007). Standish Group Report: *There's less development chaos today*. Available from: http://www.sdtimes.com/link/30247

Salo, O. & Abrahamsson, P. (2008). Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of extreme programming and scrum. *The Institute of Engineering and Technology*, 2(1), 58-64. doi:10.1049/iet-sen:20070038.

Siviy, J., Penn, M. L. & Harper, E. (2005). *Relationships between CMMI and Six Sigma*. Available from: http://www.sei.cmu.edu/reports/05tn005.pdf (accessed 01/03/2010)

SEI. (2006). *CMMI for Development, Version 2.1.* Pittsburgh: Carnegie Mellon University. Available from: http://www.sei.cmu.edu/reports/06tr008.pdf (accessed 01/03/2010)

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(2007), 883–895. doi:10.1016/j.jss.2006.09.008

The Standish Group. (1995). *Chaos Report*. Retrieved from: http://www.projectsmart.co.uk/docs/chaos-report.pdf

The Standish Group. (2009). *Standish Newsroom – Chaos 2009*. Retrieved from: http://www1.standishgroup.com/newsroom/chaos_2009.php

SurveyMonkey (2010). SurveyMonkey home page. Retrieved from: http://www.surveymonkey.com/

Szalvay, V. (2008). An introduction to agile software development. Retrieved from: http://www.danube.com/system/files/WP_Intro_to_Agile.pdf

Vinekar, V., Slinkman, C. W. & Nerur, S. (2006). Can agile and traditional systems development approaches coexist? An ambidextrous view. *Information Systems Management,* 23(3), 31-42*.* doi: 10.1201/1078.10580530/46108.23.3.20060601/93705.4

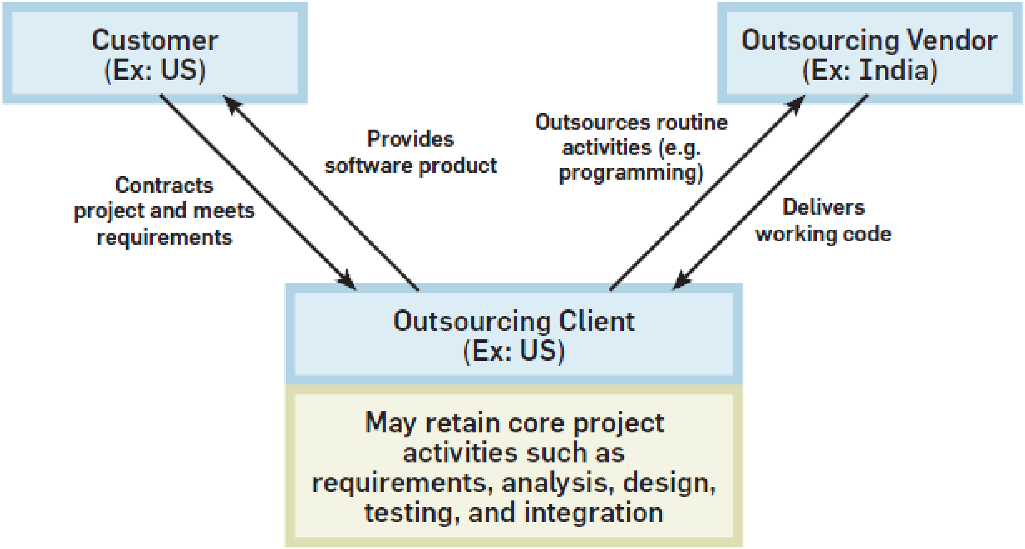
Williams , L. (2010). Agile Software Development Methodologies and Practices. *Advances in Computers*, 80, 1-44. doi:10.1016/S0065-2458(10)80001-4

Zikmund, W. G. (2003). *Business Research Methods*. 7th ed. Ohio: South Western.

9. APPENDICES

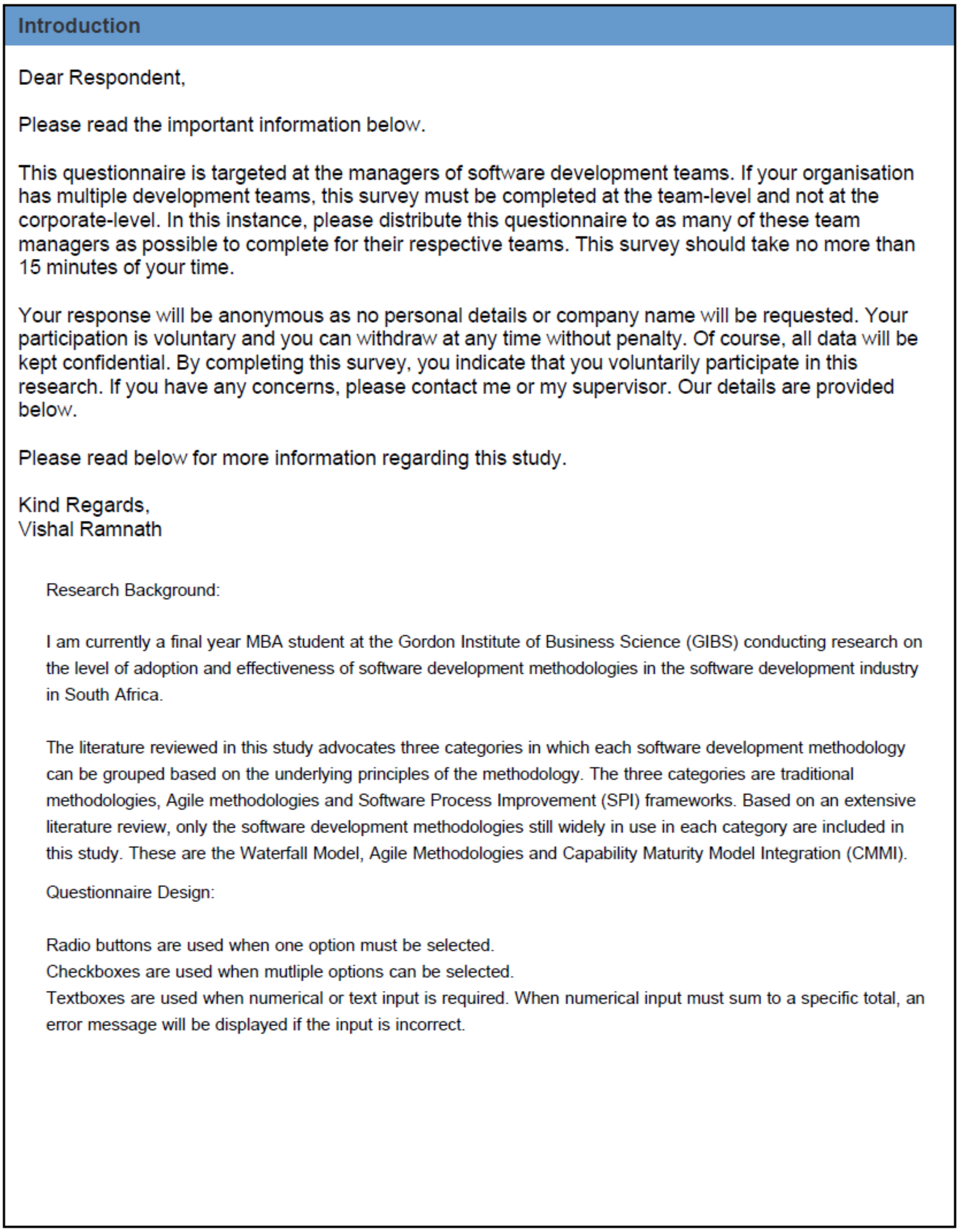
9.1. Appendix 1 – Typical Outsource Model

Figure 9.1: Typical Outsource Model


Customer
(Ex: US)
Outsourcing Vendor
(Ex: India)
Provides software product
Outsources routine activities (e.g. programming)
Contracts project and meets requirements
Delivers working code
Outsourcing Client
(Ex: US)
May retain core project activities such as requirements, analysis, design, testing, and integration

Source: *Batra (2009)*

9.2. Appendix 2 – Questionnaire



Introduction

Dear Respondent,

Please read the important information below.

This questionnaire is targeted at the managers of software development teams. If your organisation has multiple development teams, this survey must be completed at the team-level and not at the corporate-level. In this instance, please distribute this questionnaire to as many of these team managers as possible to complete for their respective teams. This survey should take no more than 15 minutes of your time.

Your response will be anonymous as no personal details or company name will be requested. Your participation is voluntary and you can withdraw at any time without penalty. Of course, all data will be kept confidential. By completing this survey, you indicate that you voluntarily participate in this research. If you have any concerns, please contact me or my supervisor. Our details are provided below.

Please read below for more information regarding this study.

Kind Regards,
Vishal Ramnath

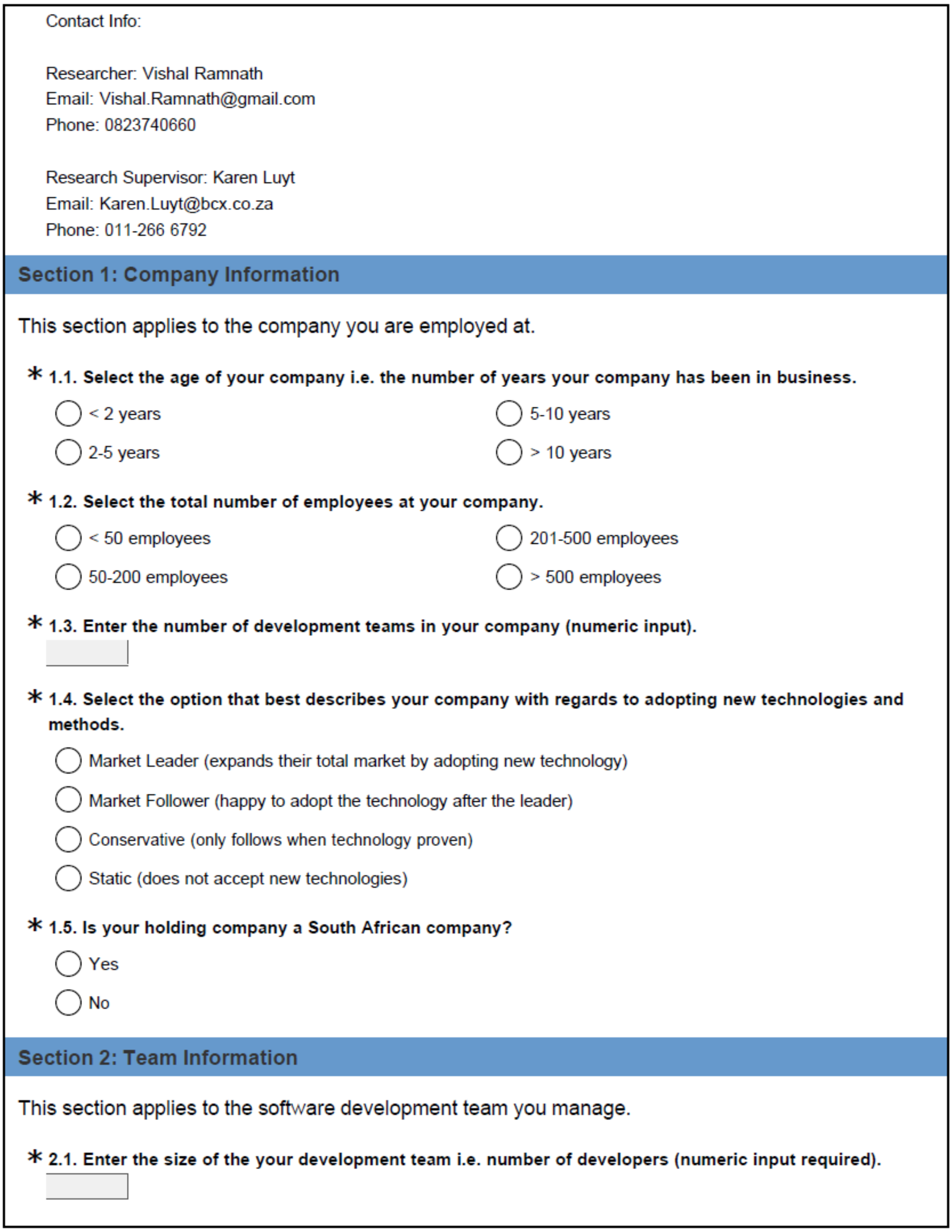
Research Background:

I am currently a final year MBA student at the Gordon Institute of Business Science (GIBS) conducting research on the level of adoption and effectiveness of software development methodologies in the software development industry in South Africa.

The literature reviewed in this study advocates three categories in which each software development methodology can be grouped based on the underlying principles of the methodology. The three categories are traditional methodologies, Agile methodologies and Software Process Improvement (SPI) frameworks. Based on an extensive literature review, only the software development methodologies still widely in use in each category are included in this study. These are the Waterfall Model, Agile Methodologies and Capability Maturity Model Integration (CMMI).

Questionnaire Design:

Radio buttons are used when one option must be selected.
Checkboxes are used when mutliple options can be selected.
Textboxes are used when numerical or text input is required. When numerical input must sum to a specific total, an error message will be displayed if the input is incorrect.



Contact Info:

Researcher: Vishal Ramnath
Email: Vishal.Ramnath@gmail.com
Phone: 0823740660

Research Supervisor: Karen Luyt
Email: Karen.Luyt@bcx.co.za
Phone: 011-266 6792

Section 1: Company Information

This section applies to the company you are employed at.

*** 1.1. Select the age of your company i.e. the number of years your company has been in business.**

○ < 2 years ○ 5-10 years
○ 2-5 years ○ > 10 years

*** 1.2. Select the total number of employees at your company.**

○ < 50 employees ○ 201-500 employees
○ 50-200 employees ○ > 500 employees

*** 1.3. Enter the number of development teams in your company (numeric input).**

*** 1.4. Select the option that best describes your company with regards to adopting new technologies and methods.**

○ Market Leader (expands their total market by adopting new technology)
○ Market Follower (happy to adopt the technology after the leader)
○ Conservative (only follows when technology proven)
○ Static (does not accept new technologies)

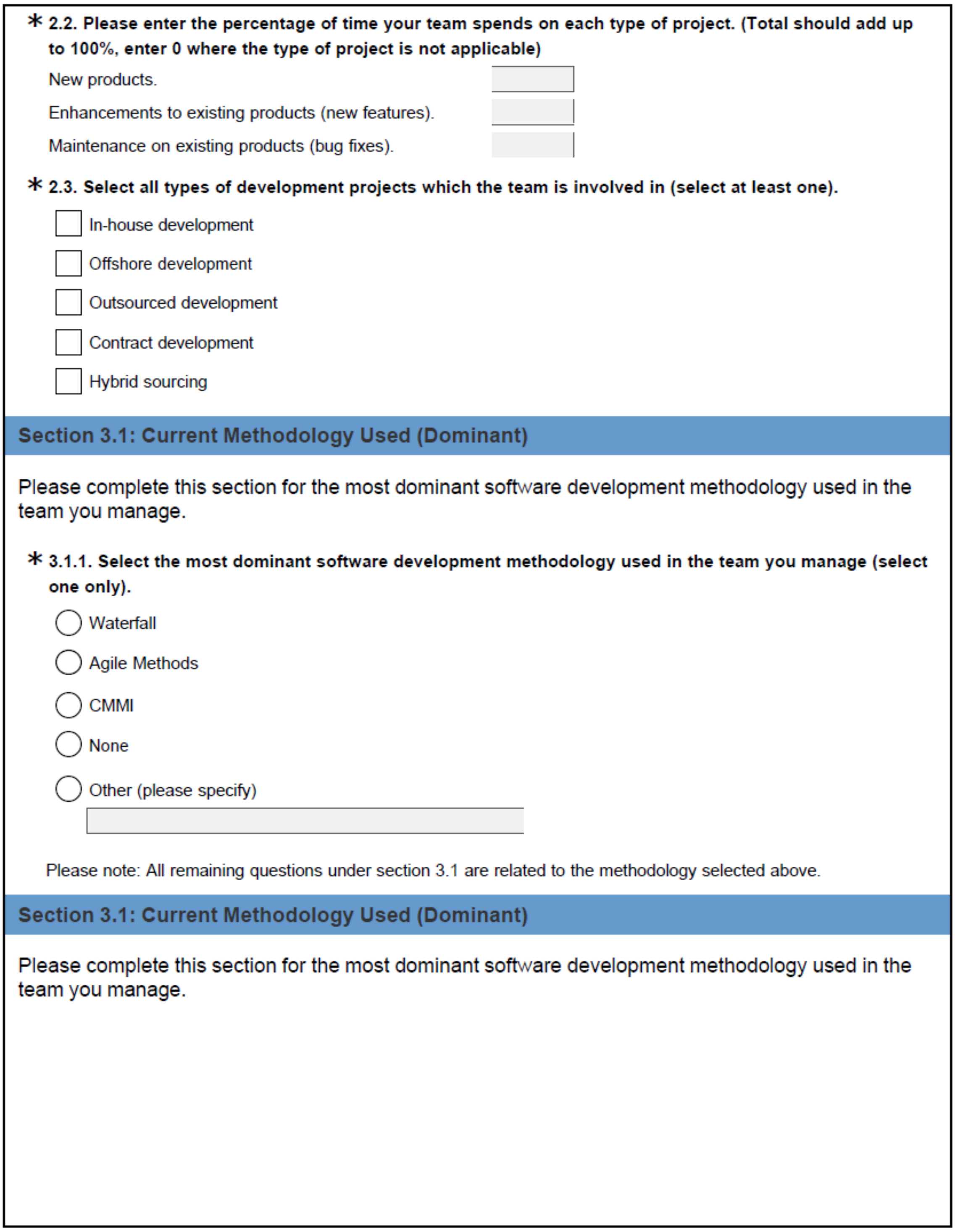
*** 1.5. Is your holding company a South African company?**

○ Yes
○ No

Section 2: Team Information

This section applies to the software development team you manage.

*** 2.1. Enter the size of the your development team i.e. number of developers (numeric input required).**



* **2.2. Please enter the percentage of time your team spends on each type of project. (Total should add up to 100%, enter 0 where the type of project is not applicable)**

New products.

Enhancements to existing products (new features).

Maintenance on existing products (bug fixes).

* **2.3. Select all types of development projects which the team is involved in (select at least one).**

□ In-house development

□ Offshore development

□ Outsourced development

□ Contract development

□ Hybrid sourcing

Section 3.1: Current Methodology Used (Dominant)

Please complete this section for the most dominant software development methodology used in the team you manage.

* **3.1.1. Select the most dominant software development methodology used in the team you manage (select one only).**

○ Waterfall

○ Agile Methods

○ CMMI

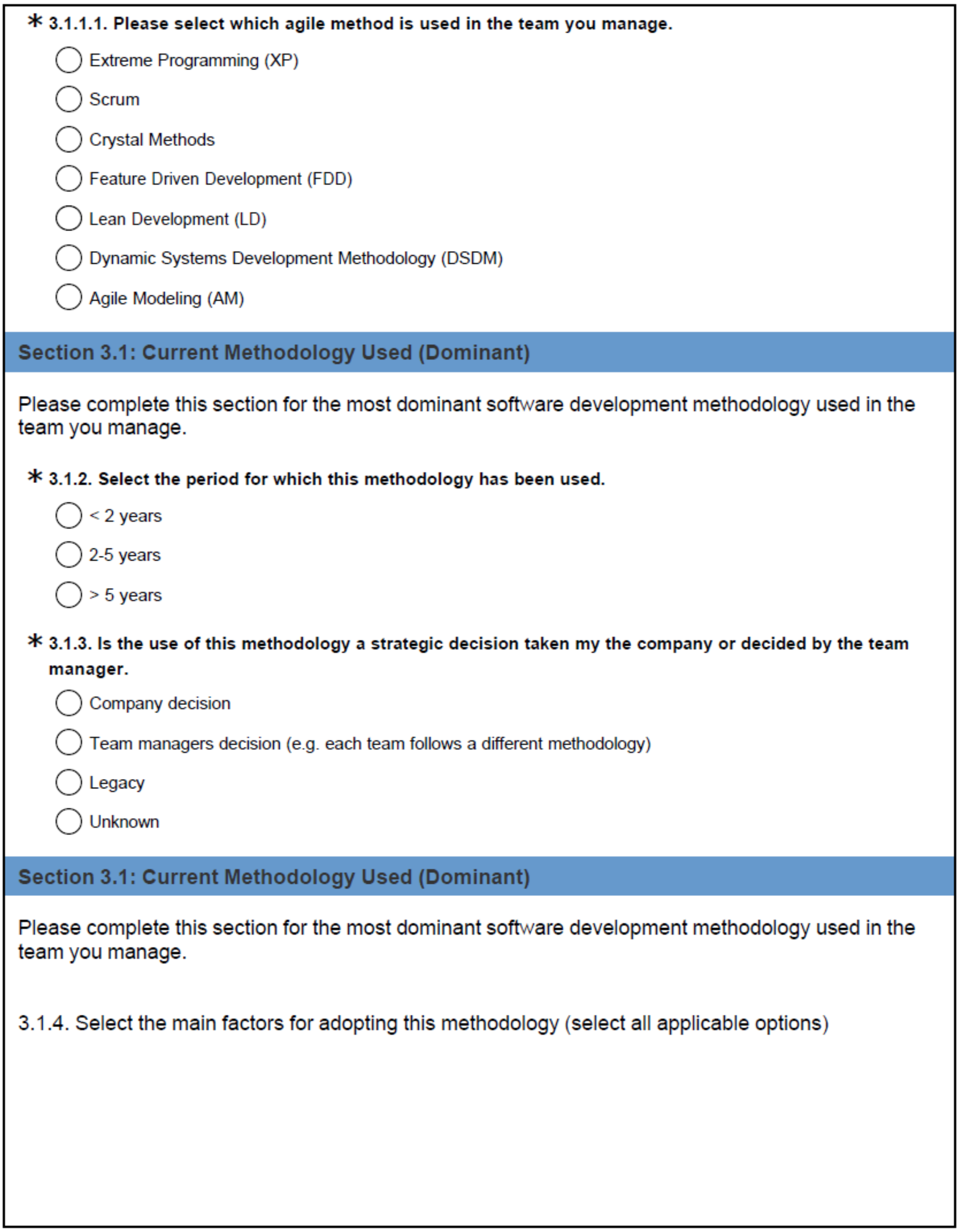
○ None

○ Other (please specify)

Please note: All remaining questions under section 3.1 are related to the methodology selected above.

Section 3.1: Current Methodology Used (Dominant)

Please complete this section for the most dominant software development methodology used in the team you manage.



* 3.1.1.1. Please select which agile method is used in the team you manage.

○ Extreme Programming (XP)

○ Scrum

○ Crystal Methods

○ Feature Driven Development (FDD)

○ Lean Development (LD)

○ Dynamic Systems Development Methodology (DSDM)

○ Agile Modeling (AM)

Section 3.1: Current Methodology Used (Dominant)

Please complete this section for the most dominant software development methodology used in the team you manage.

* 3.1.2. Select the period for which this methodology has been used.

○ < 2 years

○ 2-5 years

○ > 5 years

* 3.1.3. Is the use of this methodology a strategic decision taken my the company or decided by the team manager.

○ Company decision

○ Team managers decision (e.g. each team follows a different methodology)

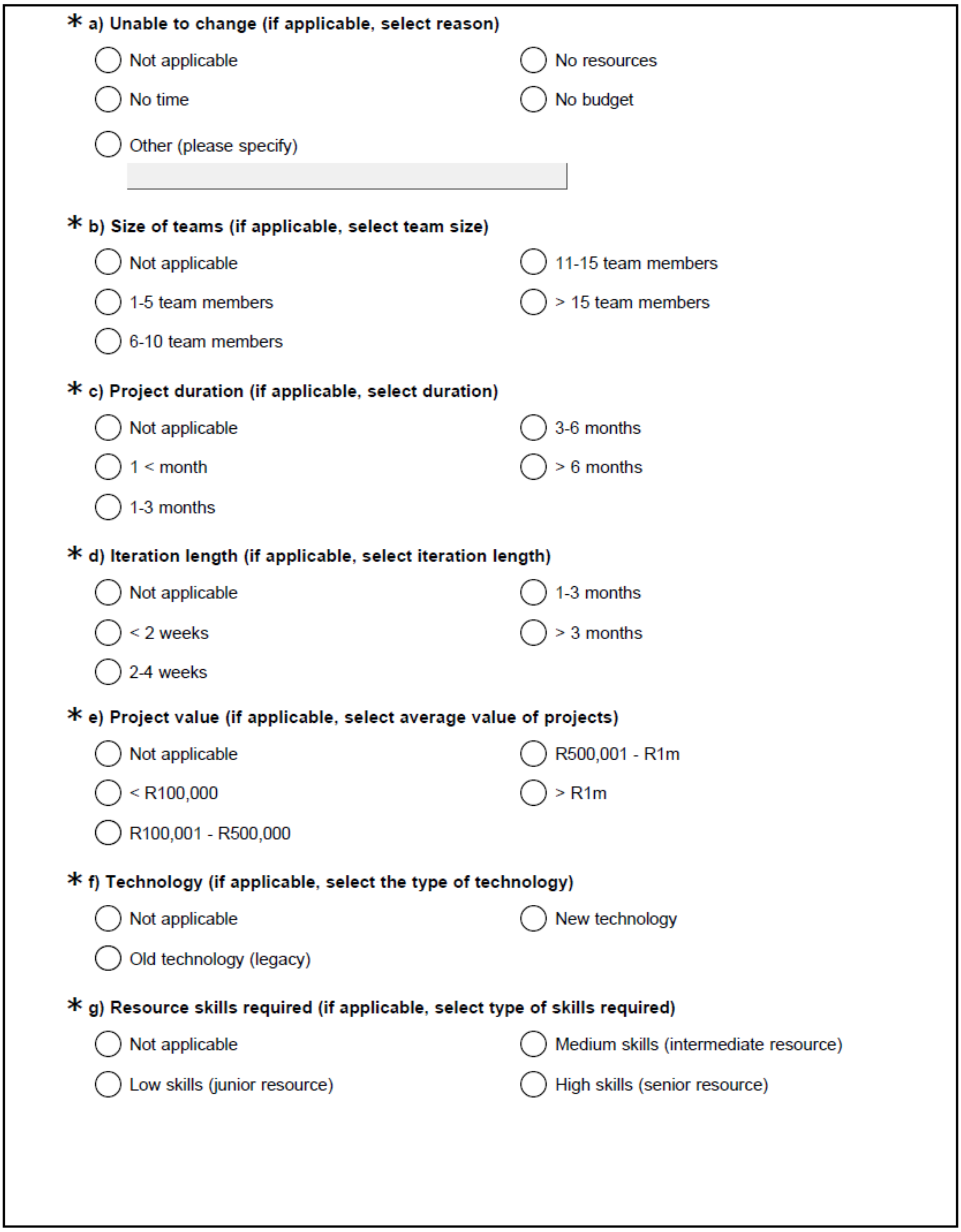
○ Legacy

○ Unknown

Section 3.1: Current Methodology Used (Dominant)

Please complete this section for the most dominant software development methodology used in the team you manage.

3.1.4. Select the main factors for adopting this methodology (select all applicable options)



*** a) Unable to change (if applicable, select reason)**

○ Not applicable

○ No time

○ Other (please specify)

○ No resources

○ No budget

*** b) Size of teams (if applicable, select team size)**

○ Not applicable

○ 1-5 team members

○ 6-10 team members

○ 11-15 team members

○ > 15 team members

*** c) Project duration (if applicable, select duration)**

○ Not applicable

○ 1 < month

○ 1-3 months

○ 3-6 months

○ > 6 months

*** d) Iteration length (if applicable, select iteration length)**

○ Not applicable

○ < 2 weeks

○ 2-4 weeks

○ 1-3 months

○ > 3 months

*** e) Project value (if applicable, select average value of projects)**

○ Not applicable

○ < R100,000

○ R100,001 - R500,000

○ R500,001 - R1m

○ > R1m

*** f) Technology (if applicable, select the type of technology)**

○ Not applicable

○ Old technology (legacy)

○ New technology

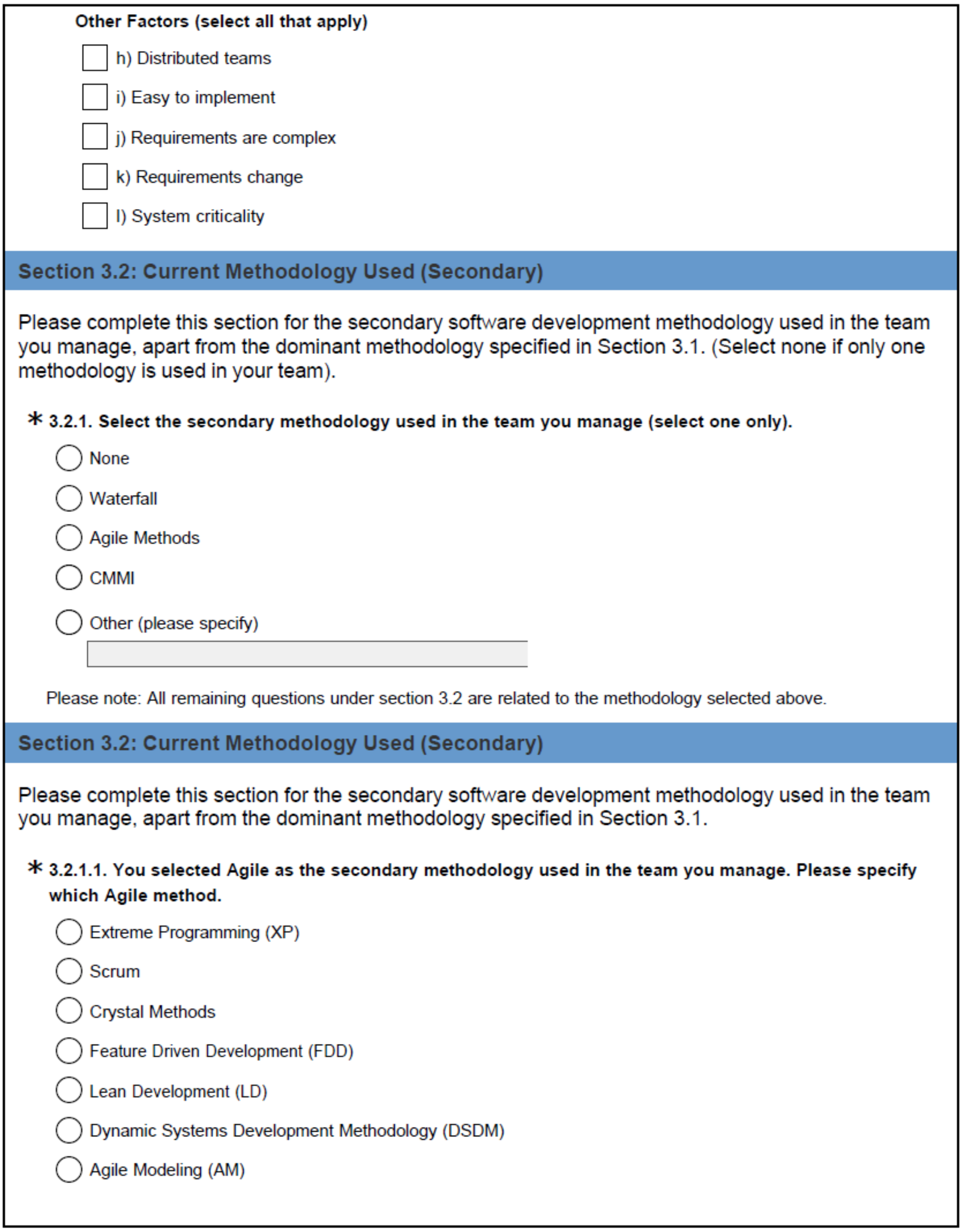
*** g) Resource skills required (if applicable, select type of skills required)**

○ Not applicable

○ Low skills (junior resource)

○ Medium skills (intermediate resource)

○ High skills (senior resource)



Other Factors (select all that apply)

□ h) Distributed teams

□ i) Easy to implement

□ j) Requirements are complex

□ k) Requirements change

□ l) System criticality

Section 3.2: Current Methodology Used (Secondary)

Please complete this section for the secondary software development methodology used in the team you manage, apart from the dominant methodology specified in Section 3.1. (Select none if only one methodology is used in your team).

*** 3.2.1. Select the secondary methodology used in the team you manage (select one only).**

○ None

○ Waterfall

○ Agile Methods

○ CMMI

○ Other (please specify)

Please note: All remaining questions under section 3.2 are related to the methodology selected above.

Section 3.2: Current Methodology Used (Secondary)

Please complete this section for the secondary software development methodology used in the team you manage, apart from the dominant methodology specified in Section 3.1.

*** 3.2.1.1. You selected Agile as the secondary methodology used in the team you manage. Please specify which Agile method.**

○ Extreme Programming (XP)

○ Scrum

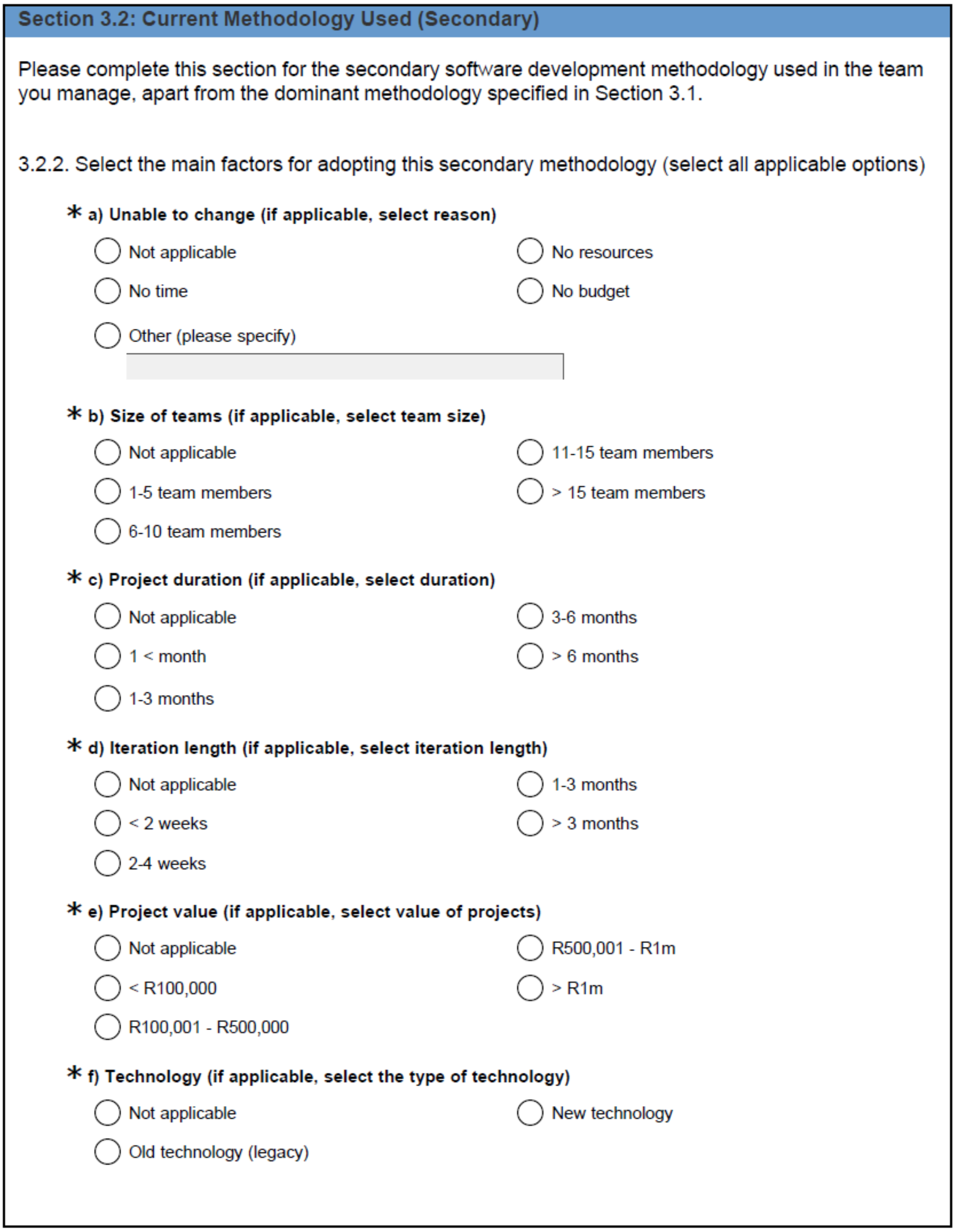
○ Crystal Methods

○ Feature Driven Development (FDD)

○ Lean Development (LD)

○ Dynamic Systems Development Methodology (DSDM)

○ Agile Modeling (AM)



Section 3.2: Current Methodology Used (Secondary)

Please complete this section for the secondary software development methodology used in the team you manage, apart from the dominant methodology specified in Section 3.1.

3.2.2. Select the main factors for adopting this secondary methodology (select all applicable options)

*** a) Unable to change (if applicable, select reason)**

○ Not applicable
○ No time
○ Other (please specify)
○ No resources
○ No budget

*** b) Size of teams (if applicable, select team size)**

○ Not applicable
○ 1-5 team members
○ 6-10 team members
○ 11-15 team members
○ > 15 team members

*** c) Project duration (if applicable, select duration)**

○ Not applicable
○ 1 < month
○ 1-3 months
○ 3-6 months
○ > 6 months

*** d) Iteration length (if applicable, select iteration length)**

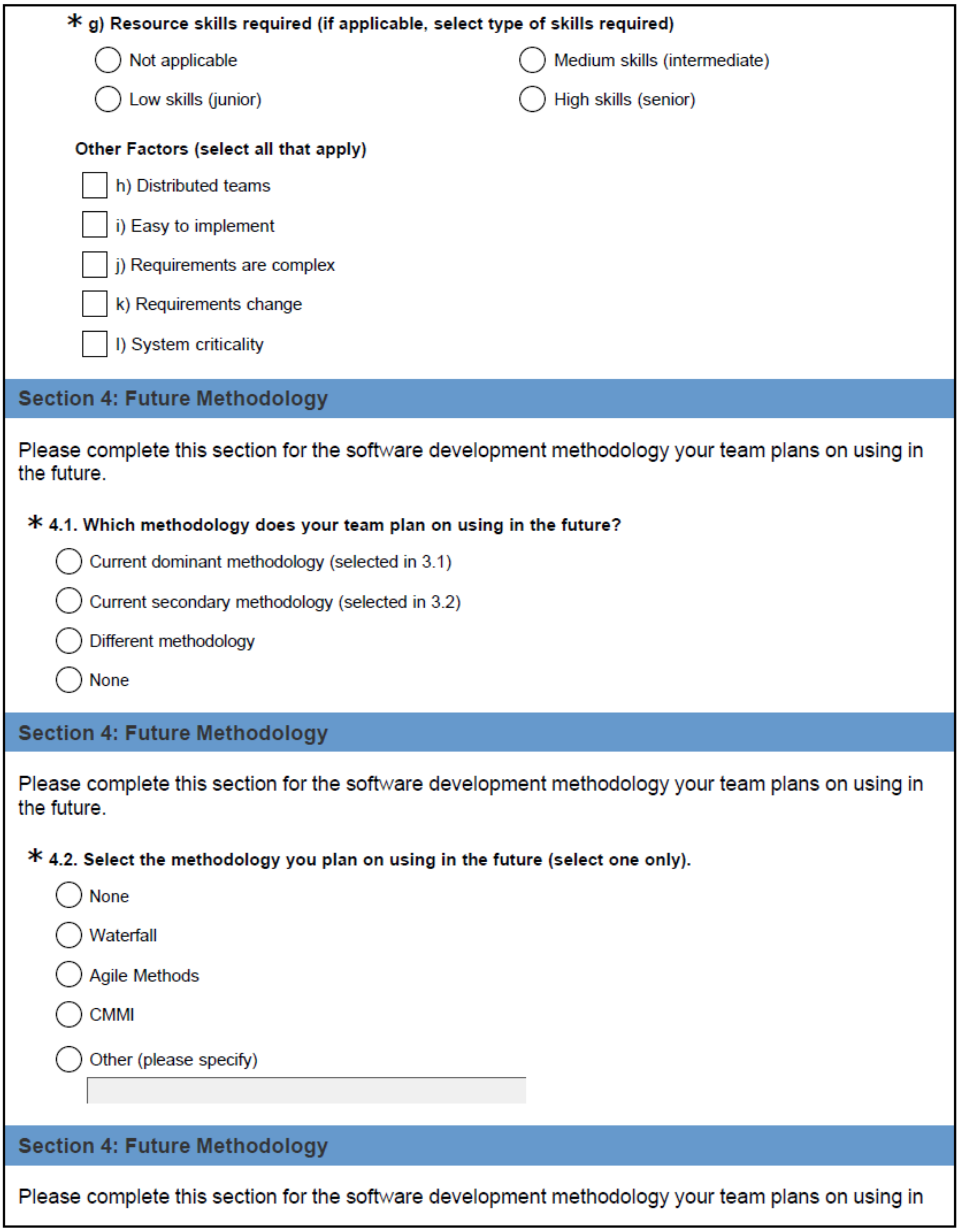
○ Not applicable
○ < 2 weeks
○ 2-4 weeks
○ 1-3 months
○ > 3 months

*** e) Project value (if applicable, select value of projects)**

○ Not applicable
○ < R100,000
○ R100,001 - R500,000
○ R500,001 - R1m
○ > R1m

*** f) Technology (if applicable, select the type of technology)**

○ Not applicable
○ Old technology (legacy)
○ New technology



* g) Resource skills required (if applicable, select type of skills required)

○ Not applicable ○ Medium skills (intermediate)

○ Low skills (junior) ○ High skills (senior)

Other Factors (select all that apply)

□ h) Distributed teams

□ i) Easy to implement

□ j) Requirements are complex

□ k) Requirements change

□ l) System criticality

Section 4: Future Methodology

Please complete this section for the software development methodology your team plans on using in the future.

* 4.1. Which methodology does your team plan on using in the future?

○ Current dominant methodology (selected in 3.1)

○ Current secondary methodology (selected in 3.2)

○ Different methodology

○ None

Section 4: Future Methodology

Please complete this section for the software development methodology your team plans on using in the future.

* 4.2. Select the methodology you plan on using in the future (select one only).

○ None

○ Waterfall

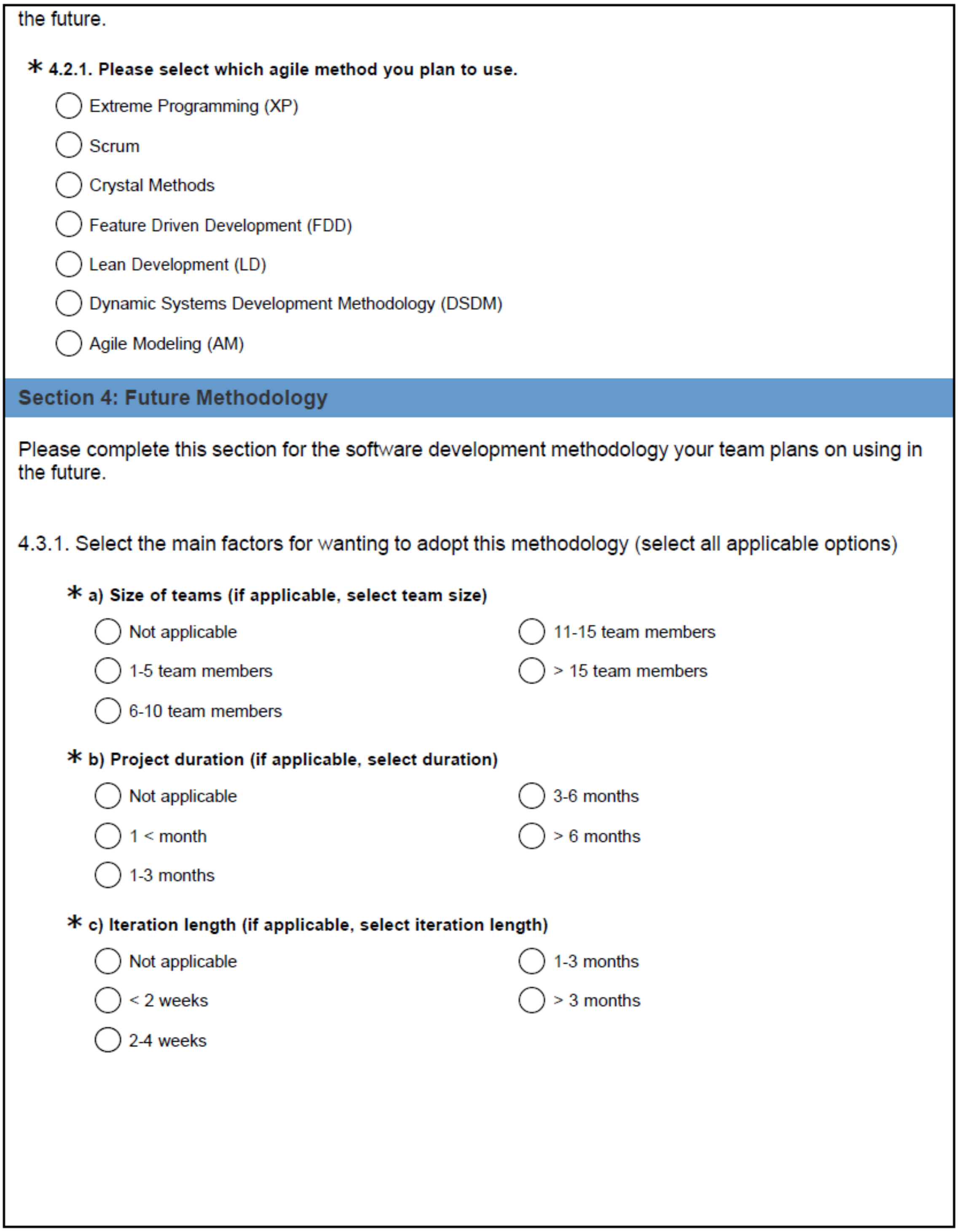
○ Agile Methods

○ CMMI

○ Other (please specify)

Section 4: Future Methodology

Please complete this section for the software development methodology your team plans on using in



the future.

*** 4.2.1. Please select which agile method you plan to use.**

○ Extreme Programming (XP)

○ Scrum

○ Crystal Methods

○ Feature Driven Development (FDD)

○ Lean Development (LD)

○ Dynamic Systems Development Methodology (DSDM)

○ Agile Modeling (AM)

Section 4: Future Methodology

Please complete this section for the software development methodology your team plans on using in the future.

4.3.1. Select the main factors for wanting to adopt this methodology (select all applicable options)

*** a) Size of teams (if applicable, select team size)**

○ Not applicable

○ 1-5 team members

○ 6-10 team members

○ 11-15 team members

○ > 15 team members

*** b) Project duration (if applicable, select duration)**

○ Not applicable

○ 1 < month

○ 1-3 months

○ 3-6 months

○ > 6 months

*** c) Iteration length (if applicable, select iteration length)**

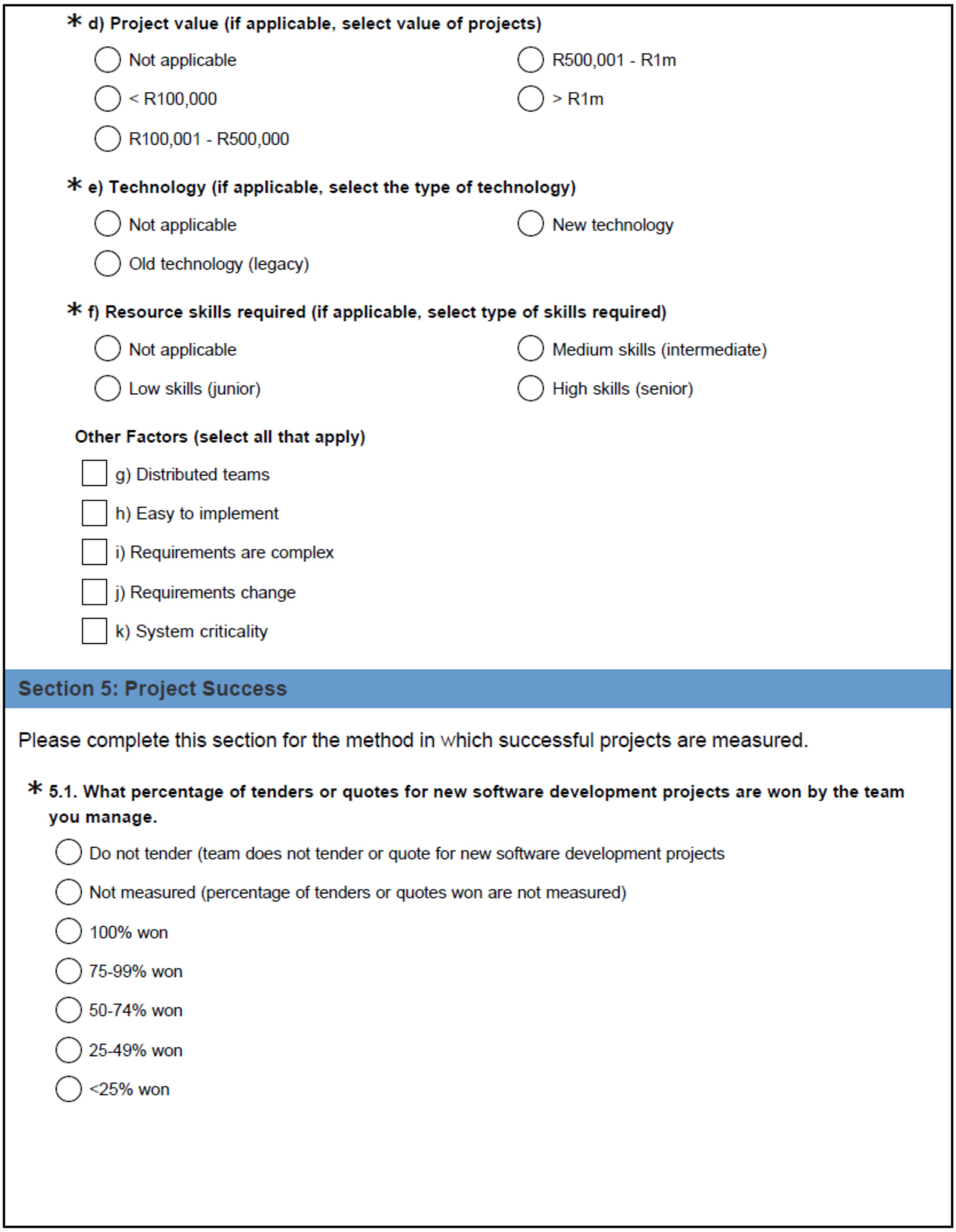
○ Not applicable

○ < 2 weeks

○ 2-4 weeks

○ 1-3 months

○ > 3 months



∗ d) Project value (if applicable, select value of projects)

○ Not applicable
○ < R100,000
○ R100,001 - R500,000
○ R500,001 - R1m
○ > R1m

∗ e) Technology (if applicable, select the type of technology)

○ Not applicable
○ Old technology (legacy)
○ New technology

∗ f) Resource skills required (if applicable, select type of skills required)

○ Not applicable
○ Low skills (junior)
○ Medium skills (intermediate)
○ High skills (senior)

Other Factors (select all that apply)

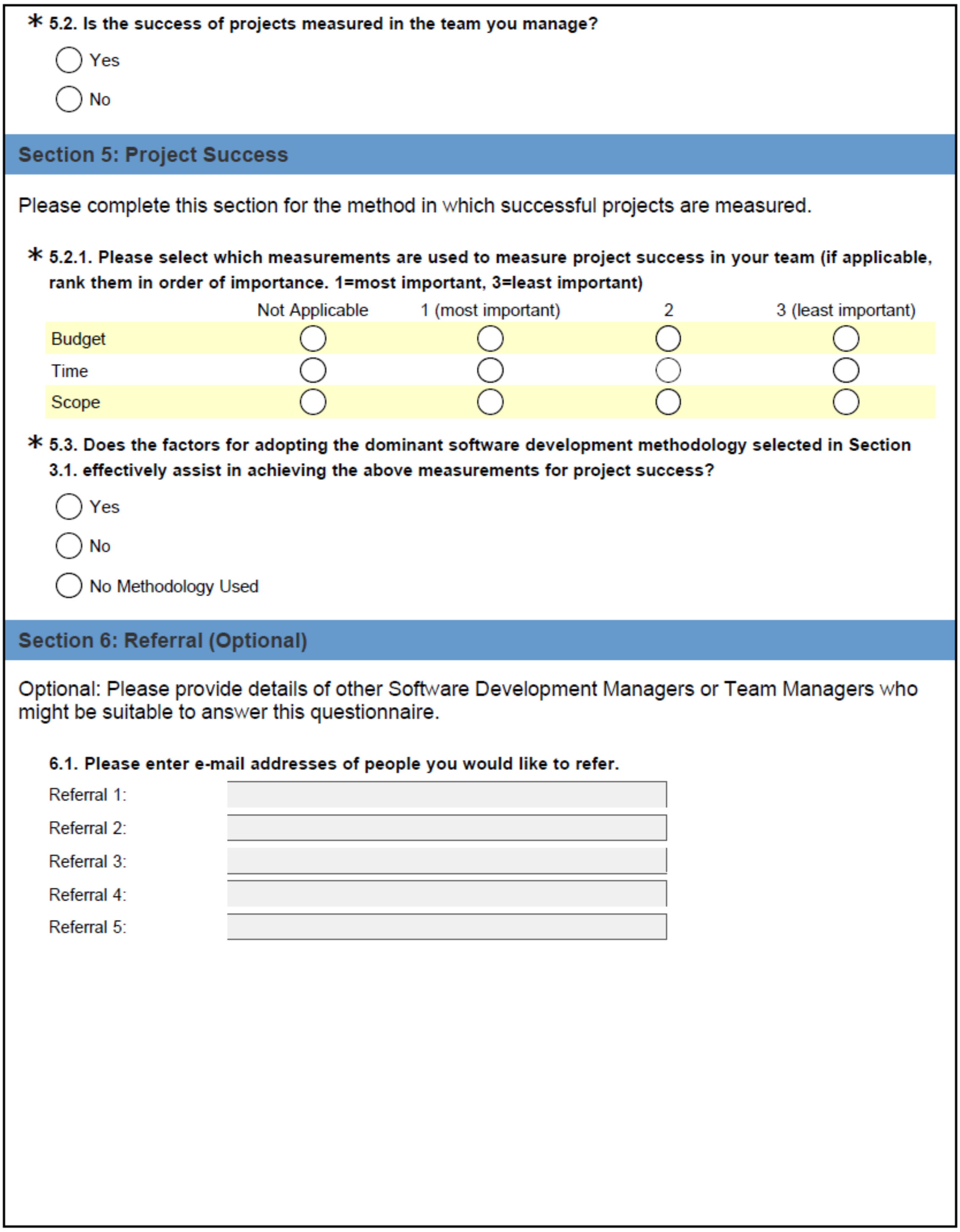
□ g) Distributed teams
□ h) Easy to implement
□ i) Requirements are complex
□ j) Requirements change
□ k) System criticality

Section 5: Project Success

Please complete this section for the method in which successful projects are measured.

∗ 5.1. What percentage of tenders or quotes for new software development projects are won by the team you manage.

○ Do not tender (team does not tender or quote for new software development projects
○ Not measured (percentage of tenders or quotes won are not measured)
○ 100% won
○ 75-99% won
○ 50-74% won
○ 25-49% won
○ <25% won



*** 5.2. Is the success of projects measured in the team you manage?**

○ Yes

○ No

Section 5: Project Success

Please complete this section for the method in which successful projects are measured.

*** 5.2.1. Please select which measurements are used to measure project success in your team (if applicable, rank them in order of importance. 1=most important, 3=least important)**

	Not Applicable	1 (most important)	2	3 (least important)
Budget	○	○	○	○
Time	○	○	○	○
Scope	○	○	○	○

*** 5.3. Does the factors for adopting the dominant software development methodology selected in Section 3.1. effectively assist in achieving the above measurements for project success?**

○ Yes

○ No

○ No Methodology Used

Section 6: Referral (Optional)

Optional: Please provide details of other Software Development Managers or Team Managers who might be suitable to answer this questionnaire.

6.1. Please enter e-mail addresses of people you would like to refer.

Referral 1:

Referral 2:

Referral 3:

Referral 4:

Referral 5: