

The development of a hybrid agile project management methodology

J. Grey

**Thesis submitted for the degree Doctor of Philosophy in Computer Science at the
Potchefstroom Campus of the North-West University**

Promoter: Prof. H.M. Huisman

June 2011

ABSTRACT

KEYWORDS: System development methodology (SDM); agile system development methodology (ASDM); project management methodology (PMM); Project IN Controlled Environment (PRINCE2); Project Management Body of Knowledge (PMBOK); Agile Project Management Methodology (APMM).

The aim of this study is to investigate whether a combination of agile system development methodologies (ASDMs) and project management methodologies (PMMs) can be used to develop a hybrid APMM that will have the ability to deliver information technology (IT) projects successfully in a constantly changing business and project environment.

To achieve this objective, a literature review was conducted on the relatively well-established ASDMs by firstly defining a SDM and an ASDM. Each ASDM and its effectiveness are described, after which ASDMs in general are evaluated by considering their area of application, advantages and disadvantages. A comparison is then done of the seven different ASDMs using the four elements of an SDM (Huisman & livari, 2006:32) to emphasise some of the main similarities and differences amongst the different ASDMs. The seven ASDMs investigated in this study are Dynamic System Development Methodology, Scrum, Extreme Programming, Feature Driven Development, Crystal ASDMs – Crystal Clear and Crystal Orange in particular, Adaptive Software Development and Lean Development.

A literature review was also conducted on two structured and relatively well-established PMMs, PMBOK and PRINCE2, and a relatively new PMM called Agile Project Management. Each PMM is evaluated by considering their area of application, advantages, disadvantages and integration with other methodologies, after which a comparison is made of the different PMMs.

The research was conducted by following a mixed methods research plan, which included the mixed methods research paradigm (combination of the interpretive research paradigm and the positivistic research paradigm), research methods (design science, case study and survey), quantitative and qualitative data-collection techniques (interviews and questionnaires), and data-analysis techniques (cross-case and statistical).

The reasons that projects fail and critical project success factors were studied and summarised to form the *critical project success criteria*, which were used to create the *agile project success*

criteria. The ASDM best practice and PMM best practice frameworks were created by identifying whether a certain ASDM or PMM would satisfy a specific agile project success factor (APSF) of the *agile project success criteria*. The findings of each APSF in the respective frameworks were used as a foundation to develop a hybrid APMM (ver. 0) that would address the *agile project success criteria*. The hybrid APMM (ver. 0) was developed interpretively using design science (research approach) and constructivism by combining the strengths, addressing the weaknesses and bridging the gaps identified in the frameworks.

The hybrid APMM (ver. 0) was then evaluated and improved by conducting an interpretive case study, which entailed interviewing participants from large and small organisations. Once the qualitative data collected had been analysed using cross-case analysis, the findings were incorporated in order to create an improved hybrid APMM (ver. 1).

The hybrid APMM (ver. 1) too was evaluated and improved by conducting a survey, which entailed administering questionnaires to various respondents in order to collect quantitative and qualitative data. The findings of the statistical analysis of the data were also used to improve the hybrid APMM (ver. 1), resulting in the final hybrid APMM (ver. 2).

This study demonstrates that a combination of ASDMs and PMMs can be used to develop a hybrid APMM with the ability to deliver IT projects in a constantly changing project and business environment.

OPSOMMING

SLEUTELWOORDE: Stelselontwikkelingsmetodologie (SDM); vinnig aanpasbare stelselontwikkelingsmetodologie (ASDM); projekbestuursmetodologie (PMM); vinnig aanpasbare projekbestuursmetodologie (APMM); projek in 'n gekontroleerde omgewing (PRINCE2); projekbestuurkundigheidsliggaam (PMBOK); aanpasbare projekbestuur (APM); aanpasbare projekbestuursmetodologie (APMM); vinnig aanpasbare projek suksesfaktor (APSF).

Die doel van hierdie studie was om te bepaal of 'n kombinasie van ASDM'e (vinnig aanpasbare stelselontwikkelingsmetodologieë) en PMM'e (projekbestuursmetodologieë) gebruik kan word om 'n hibried APMM (vinnig aanpasbare projekbestuursmetodologie) te ontwikkel wat die vermoë sal hê om informasie tegnologie (IT) projekte suksesvol in 'n voortdurend veranderende projek- en besigheidsomgewing te voltooi.

Om hierdie doel te bereik, is 'n literatuurstudie gedoen oor die relatief goed gevestigde ASDM'e. Eerstens is 'n definisie van 'n SDM (stelselontwikkelingsmetodologie) en 'n ASDM (vinnig aanpasbare stelselontwikkelingsmetodologie) gegee. Hierna is elke ASDM en sy effektiwiteit beskryf, waarna ASDM'e in die algemeen geëvalueer is deur die area van toepassing, voordele, en nadele, in ag te neem. Die sewe ASDM'e is vergelyk deur die elemente van 'n SDM (Huisman & livari, 2006:32) te gebruik om die hoofooreenkomste en -verskille tussen die verskillende ASDM'e uit te lig. Die volgende sewe ASDM'e is in hierdie studie ondersoek: Dinamiese Stelselontwikkelingsmetodologie (DSDM), Skrum, Ekstreme Programming (XP), Kenmerk Gedrewe Ontwikkeling (FDD), Kristal Vinnig Aanpasbare Stelselontwikkelingsmetodologie (Crystal ASDM) – veral Kristal Helder (CC) en Kristal Oranje (CO), Aanpasbare Stelselontwikkelingsmetodologie (ASD) en Eenvoudige en Aanpasbare Ontwikkeling (LD).

'n Literatuurstudie is ook gedoen oor twee gestruktureerde en relatief goed gevestigde PMM'e; PMBOK (projekbestuurkundigheidsliggaam) en PRINCE2 (projek in 'n gekontroleerde omgewing), en 'n relatief nuwe PMM wat APM (aanpasbare projekbestuur) genoem word. Elke PMM is geëvalueer deur die area van toepassing, voordele, nadele, en die vermoë om met ander metodologieë geïntegreer te word te verduidelik. Hierna is 'n vergelyking tussen die verskillende PMM'e getref.

Die navorsing is gedoen deur 'n gemengde metode navorsingsplan te volg wat die gemengde metode navorsingsparadigma (kombinasie van die interpretatiewe navorsingsparadigma en die

positivistiese navorsingsparadigma), navorsingsmetodes (ontwerpwetenskap, gevallestudie en opname), kwantitatiewe en kwalitatiewe dataversameling tegnieke (onderhoude en vraelyste), en data-analise tegnieke (kruisgevalle en statisties), insluit.

Die kritiese projek suksesfaktore asook die redes waarom projekte misluk, is bestudeer en saamgevat om *die kritiese projek sukseskriteria* te vorm wat gebruik was om die *vinnig aanpasbare projek sukseskriteria* daar te stel. Die ASDM algemeen toepaslike en PMM algemeen toepaslike raamwerke is ontwerp deur te bepaal of 'n sekere ASDM of PMM aan 'n spesifieke vinnig aanpasbare projek suksesfaktor (APSF) in die *vinnig aanpasbare projek sukseskriteria* voldoen. Die bevindinge van elke APSF in die onderskeidelike raamwerke is gebruik as 'n fondasie waarop die hibried APMM (weergawe 0) ontwikkel is wat die *vinnig aanpasbare projek sukseskriteria* aanspreek. Die hibried APMM (weergawe 0) is interpretatief ontwikkel deur gebruik te maak van ontwerpwetenskap (navorsings metode) en konstruktivisme deur die sterk punte te kombineer, die swak punte te adresseer, en die gapings wat in die raamwerke geïdentifiseer is, te oorbrug.

Die hibried APMM (weergawe 0) is geëvalueer en verbeter deur 'n interpretatiewe gevallestudie te voltooi. Onderhoude is met deelnemers gevoer ten einde 'n verbeterde hibried APMM (weergawe 1) te ontwikkel deur die bevindinge te inkorporeer, nadat die kwalitatiewe data wat versamel is, geanaliseer is deur middel van oorkruis analise.

Die hibried APMM (weergawe 1) is verder geëvalueer en verbeter deur 'n ondersoek te doen waar vraelyste aan verskillende deelnemers gestuur is om kwalitatiewe- en kwantitatiewe data in te samel wat dan statisties geanaliseer is sodat die bevindinge by die finale verbeterde hibried APMM (weergawe 2) geïnkorporeer kon word.

Hierdie studie het suksesvol aangetoon dat 'n kombinasie van ASDM'e en PMM'e gebruik kan word om 'n hibried APMM te ontwikkel wat die vermoë het om IT projekte in 'n voortdurend veranderende projek en besigheidsomgewing te voltooi.

ACKNOWLEDGEMENTS

“Any giant will fall if you use the stones God give you correctly”

I specially thank my beautiful wife for her encouraging words, prayers and ongoing support during the past few years. Another thank-you goes out to my parents for the support and prayers since I was young.

A special thank-you goes out to Prof. Magda Huisman, for her insight and guidance in the work that I have been doing, and for always assuring me that I have done good work.

I would also like to thank the Statistical Consultation Service at the Potchefstroom Campus of the North-West University, for the review and approval of the statistical results of this study. Another thank-you goes out to the editor for revising grammar and spelling.

Most importantly, all the praise, glory and honour go to Jesus Christ alone who gave me the perseverance and the opportunity to kill this giant with the stones He gave me.

TABLE OF CONTENTS

ABSTRACT	I
OPSOMMING	III
ACKNOWLEDGEMENTS	V
TABLE OF CONTENTS	VII
LIST OF FIGURES	XIII
LIST OF TABLES.....	XV
LIST OF ACRONYMS	XVII

CHAPTER 1 INTRODUCTION

1.1 INTRODUCTION	1
1.2 PROBLEM STATEMENT AND SUBSTANTIATION.....	1
1.3 RESEARCH QUESTION	3
1.4 RESEARCH OBJECTIVES	3
1.5 METHOD OF INVESTIGATION	3
1.6 STRUCTURE OF THE THESIS.....	4
1.7 SUMMARY	6

CHAPTER 2 AGILE SYSTEM DEVELOPMENT METHODOLOGIES

2.1 INTRODUCTION	7
2.2 DEFINITION OF “SYSTEM DEVELOPMENT METHODOLOGY”	7
2.3 THE AGILE SYSTEM DEVELOPMENT METHODOLOGY	9
2.3.1 Agile Software Development Manifesto.....	9
2.3.2 Definition of “agile system development methodology”	10
2.3.3 Characteristics of an agile system development methodology	11
2.4 THE SEVEN AGILE SYSTEM DEVELOPMENT METHODOLOGIES	12
2.4.1 Dynamic System Development Methodology.....	13
2.4.2 Scrum	19
2.4.3 Extreme Programming	23
2.4.4 Feature Driven Development	32
2.4.5 Crystal agile system development methodologies	37

2.4.6 Adaptive Software Development	43
2.4.7 Lean Development.....	49
2.5 THE EFFECTIVENESS AND ACCEPTANCE OF AGILE SYSTEM DEVELOPMENT METHODOLOGIES IN PRACTICE.....	56
2.6 COMPARISON OF THE SEVEN AGILE SYSTEM DEVELOPMENT METHODOLOGIES.....	64
2.7 SUMMARY	65

CHAPTER 3

PROJECT MANAGEMENT METHODOLOGIES

3.1 INTRODUCTION	69
3.2 DEFINITION OF “PROJECT MANAGEMENT”	69
3.3 PROJECT MANAGEMENT BODY OF KNOWLEDGE	74
3.3.1 PMBOK process groups	74
3.3.1.1 Initiation process	75
3.3.1.2 Planning process.....	76
3.3.1.3 Execution process.....	77
3.3.1.4 Monitoring and controlling process	77
3.3.1.5 Closing process.....	78
3.3.2 PMBOK's nine knowledge areas	78
3.3.2.1 Project integration management	80
3.3.2.2 Project scope management.....	81
3.3.2.3 Project time management.....	82
3.3.2.4 Project cost management.....	84
3.3.2.5 Project quality management	84
3.3.2.6 Project human resource management	85
3.3.2.7 Project communications management	86
3.3.2.8 Project risk management.....	87
3.3.2.9 Project procurement management.....	88
3.3.3 Evaluation of PMBOK.....	89
3.4 PROJECT IN CONTROLLED ENVIRONMENT	94
3.4.1 PRINCE2 structure	99
3.4.1.1 PRINCE2 project environment.....	100
3.4.1.2 PRINCE2 principles.....	101
3.4.1.3 PRINCE2 themes.....	102
3.4.2 PRINCE2 process model.....	104

3.4.2.1 Starting up a project	105
3.4.2.2 Directing a project	106
3.4.2.3 Initiating a project	108
3.4.2.4 Controlling a stage	110
3.4.2.5 Managing product delivery	111
3.4.2.6 Managing a stage boundary	112
3.4.2.7 Closing a project	114
3.4.3 Evaluation of PRINCE2.....	116
3.5 AGILE PROJECT MANAGEMENT	121
3.5.1 Agile Project Management values and principles	122
3.5.2 Agile enterprise framework.....	123
3.5.3 Agile Project Management Model and Delivery Framework	125
3.5.3.1 Envision	126
3.5.3.2 Speculate	128
3.5.3.3 Explore	129
3.5.3.4 Adapt	131
3.5.3.5 Close	132
3.5.4 Evaluation of Agile Project Management.....	133
3.6 COMPARISON OF PMBOK, PRINCE2 AND AGILE PROJECT MANAGEMENT ...	136
3.7 SUMMARY	141

CHAPTER 4

RESEARCH DESIGN

4.1 INTRODUCTION	143
4.2 RESEARCH PARADIGM	143
4.2.1 Positivism	144
4.2.2 Interpretivism.....	145
4.2.3 Critical social theory	148
4.2.4 Mixed methods research paradigm.....	149
4.3 RESEARCH METHOD	154
4.3.1 Design science	154
4.3.2 Research plan.....	156
4.3.2.1 Case studies	158
4.3.2.2 Survey	163
4.4 SUMMARY	169

CHAPTER 5

DEVELOPMENT OF A HYBRID AGILE PROJECT MANAGEMENT METHODOLOGY

5.1 INTRODUCTION	171
5.2 THEORETICAL DEDUCTIONS.....	172
5.2.1 Why do projects fail?	172
5.2.2 Critical project success factors.....	174
5.2.3 Critical project success criteria.....	179
5.2.4 Agile project success criteria	187
5.2.5 Agile system development methodology best practice framework.....	192
5.2.6 Project management methodology best practice framework.....	195
5.3 DEVELOPMENT OF A HYBRID APMM	197
5.3.1 Need for a hybrid agile project management methodology	198
5.3.2 Proposed hybrid agile project management methodology	198
5.3.2.1 Pre-initiation phase.....	199
5.3.2.2 Vision and definition phase.....	200
5.3.2.3 Preparation phase	201
5.3.2.4 Collaborative development phase.....	204
5.3.2.5 Close-out and hand-over phase.....	209
5.3.2.6 Adapt, direct, monitor and control phase.....	209
5.3.2.7 Post-project maintenance and continual improvement phase	211
5.4 SUMMARY	212

CHAPTER 6

EVALUATION OF THE HYBRID AGILE PROJECT MANAGEMENT METHODOLOGY

6.1 INTRODUCTION	213
6.2 EVALUATION OF THE HYBRID APMM (VER. 0)	213
6.2.1 Content analysis.....	213
6.2.1.1 Results of case study A (large organisation)	214
6.2.1.2 Results of case study B (small organisation).....	225
6.2.2 Cross-case analysis	234
6.2.2.1 Results of cross-case analysis of case studies A and B	234
6.2.3 Discussion of cross-case analysis results and suggested improvements to the hybrid agile project management methodology (ver. 0).....	244

6.3 EVALUATION OF THE HYBRID APMM (VER. 1)	256
6.3.1 Statistical analysis	257
6.3.1.1 Results of survey.....	258
6.3.2 Discussion of statistical analysis results and suggested improvements to the hybrid agile project management methodology (ver. 1)	271
6.4 SUMMARY	273

CHAPTER 7

FINDINGS AND FUTURE WORK

7.1 INTRODUCTION	275
7.2 RESEARCH OBJECTIVES AND FINDINGS	275
7.2.1 Develop the agile project success criteria.....	275
7.2.2 Develop the agile system development methodology best practice framework	279
7.2.3 Develop the project management methodology best practice framework.....	281
7.2.4 Develop a proposed hybrid APMM (ver. 0) that would address the agile project success criteria	283
7.2.5 Develop an improved hybrid APMM (ver. 1) by conducting two case studies	284
7.2.6 Develop an improved hybrid APMM (ver. 2) by conducting a survey	284
7.3 CONTRIBUTION AND CONCLUSION OF THE STUDY: HYBRID AGILE PROJECT MANAGEMENT METHODOLOGY (VER. 2)	285
7.4 LIMITATIONS	300
7.5 FUTURE WORK	301

ANNEXURES

ANNEXURE A : INTERVIEW QUESTIONS	303
ANNEXURE B : CHANGE-REQUEST FORM	305
ANNEXURE C : QUESTIONNAIRE	307
REFERENCES	317

LIST OF FIGURES

Figure 2.1:	A historical timeline of the development of the ASDMs.....	13
Figure 2.2:	The traditional approach versus the DSDM approach	14
Figure 2.3:	The DSDM life cycle.....	15
Figure 2.4:	The Scrum life cycle	20
Figure 2.5:	The XP structure	24
Figure 2.6:	The XP life cycle	27
Figure 2.7:	The practices and main cycles of XP.....	31
Figure 2.8:	The rhythm of an XP project.....	31
Figure 2.9:	FDD processes	34
Figure 2.10:	Component assembly in FDD.....	36
Figure 2.11:	The framework of Crystal ASDMs	38
Figure 2.12:	The ASD life cycle	44
Figure 2.13:	ASD life-cycle activities	45
Figure 2.14:	A conceptual framework for ASDM acceptance.....	57
Figure 2.15:	The Agile Adoption and Implementation Model	59
Figure 3.1:	A historical timeline of project management	70
Figure 3.2:	The level of process interaction and overlap of process groups over time	75
Figure 3.3:	Project management framework.....	79
Figure 3.4:	The structure of PRINCE2.....	99
Figure 3.5:	The PRINCE2 process model	104
Figure 3.6:	The agile enterprise framework	124
Figure 3.7:	The APM delivery framework	125
Figure 3.8:	The envision and explore cycles	126
Figure 3.9:	The APM life cycle	133
Figure 4.1:	The process stages of mixed methods research	151
Figure 4.2:	Reasoning in the design science cycle.....	155
Figure 4.3:	The research plan	156
Figure 5.1:	The process followed in developing the hybrid APMM (ver. 0)	171
Figure 5.2:	The twelve APSFs.....	188
Figure 5.3:	The phases of the proposed hybrid APMM life cycle	199
Figure 5.4:	The preparation phase	202
Figure 5.5:	The release plan	203
Figure 5.6:	The collaborative development phase	205
Figure 7.1:	The proposed hybrid APMM (ver. 2) life-cycle phases.....	286
Figure 7.2:	The preparation phase	289
Figure 7.3:	The release plan	290
Figure 7.4:	The collaborative development phase	292

LIST OF TABLES

Table 2.1:	Four dimensions of 4-DAT	58
Table 2.2:	Comparison of ASDMs.....	64
Table 2.3:	Summary of ASDMs.....	66
Table 2.4:	Evaluation of ASDMs towards ASDM characteristic features and values	67
Table 3.1:	Definitions of “project”	71
Table 3.2:	Definitions of “project management”	72
Table 3.3:	Project management process group and knowledge area mapping.....	79
Table 3.4:	Comparison between PRINCE2 2005 and PRINCE2 2009	97
Table 3.5:	Mapped phases of PMM life cycles	137
Table 3.6:	PRINCE2 and APM component comparison with reference to PMBOK’s knowledge areas	140
Table 4.1:	Definitions of “mixed methods research”	150
Table 5.1:	Critical project success factor ranking	177
Table 5.2:	CHAOS Ten: Recipe for project success.....	177
Table 5.3:	CPSFs and critical failure factors according to the project phase	178
Table 5.4:	Agile project success criteria	190
Table 5.5:	ASDM best practice framework	194
Table 5.6:	Extent to which ASDMs address the agile project success criteria	195
Table 5.7:	PMM best practice framework	196
Table 5.8:	Extent to which PMMs address the agile project success criteria	197
Table 6.1:	Supporting CPSF grouping	251
Table 6.2:	One-way ANOVA results for the relevance of APSFs to agile projects	259
Table 6.3:	Sorted averages of the APSFs relevant to agile projects.....	260
Table 6.4:	Comparison of the top ten APSFs with Kim et al. and Standish Group International	261
Table 6.5:	Grouping of the remaining three CPSFs of Kim et al. under the prioritised APSFs ..	262
Table 6.6:	Grouping of the remaining seven CPSFs of Standish Group International under the prioritised APSFs.....	262
Table 6.7:	One-way ANOVA results for the extent to which the hybrid APMM (ver. 1) addresses the agile project success criteria	264
Table 6.8:	Sorted averages of APSFs addressed by the hybrid APMM (ver. 1).....	265
Table 6.9:	Results of the paired t-test	268
Table 7.1:	Supporting CPSF grouping	276
Table 7.2:	ASDM best practice framework	280
Table 7.3:	PMM best practice framework	282

LIST OF ACRONYMS

4-DAT	4-Dimensional Analytical Tool
AAIM	Agile Adoption and Implementation Model
ANOVA	Analysis Of Variance
APM	Agile Project Management
APMM	Agile Project Management Methodology
APSF	Agile Project Success Factor
APSFs	Agile Project Success Factors
ASD	Adaptive Software Development
ASDM	Agile System Development Methodology
ASDMs	Agile System Development Methodology
ASSF	Agile Software Solution Framework
CC	Crystal Clear
CMMI	Capability Maturity Model Integration
CO	Crystal Orange
COBIT	Control Objectives for Information and related Technology
CPSF	Critical Project Success Factor
CPSFs	Critical Project Success Factors
DSDM	Dynamic System Development Methodology
FDD	Feature Driven Development
IT	Information Technology
ITIL	Information Technology Infrastructure Library
JAD	Joint Application Development
LD	Lean Development
OGC	Office of Government Commerce
PMBOK	Project Management Body of Knowledge
PMI	Project Management Institute
PMM	Project Management Methodology
PMMs	Project Management Methodologies
PRINCE2	PRoject IN Controlled Environment
SDM	System Development Methodology
SDMs	System Development Methodologies
XP	Extreme Programming

CHAPTER 1

INTRODUCTION

1.1 Introduction

In this chapter, the aim of this study will be described. The problem will first be described and the necessary substantiation provided to explain the need and reasons for the study to be conducted, after which the main research question will be provided. The six main research objectives necessary to address the research question will be listed. In order to fulfil these objectives and to answer the research question the method of investigation will be explained. Lastly, the manner in which the chapters will be presented and a short explanation of each will be provided.

1.2 Problem statement and substantiation

Information technology (IT) projects are very difficult projects to manage, as they have the tendency to change owing to elements of uncertainty such as project time and budget instability, constantly changing business and user requirements, and the team's ability to respond to new expectations. Because of the evolving business environment, the requirements set by business and users change frequently and unexpectedly. Consequently, there is a demand for system development methodologies (SDMs) and project management methodologies (PMMs) with the ability to adapt to a changing project and business environment.

This study will investigate seven relatively well-established agile system development methodologies (ASDMs), namely Dynamic System Development Methodology (DSDM; 1994), Scrum (1995), Extreme Programming (XP; 1998), Feature Driven Development (FDD; 1998), Crystal Methods (specifically Crystal Clear [CC; 1999]), Adaptive Software Development (ASD; 2000) and the most recent Lean Development (LD; from 2000), which started as Lean Manufacturing.

The reason for only studying these seven ASDMs is that they are the core group that are well-established in practice as explained in the Agile Software Development Manifesto (Beck *et al.*, 2001;¹ Highsmith 2002a:6;² Lindstrom & Jeffries, 2004:44; Schuh, 2004:17). There are other SDMs with an agile flavour, such as Agile Modelling, Agile Unified Process, Essential Unified

¹ References containing no page numbers are website publications (normally in HTML or .PDF format) that contain no page numbers.

² References containing page numbers are used for referencing full text Internet articles, journal articles, newspaper articles, magazine articles and books.

Process, Open Unified Process and Velocity Tracking, which will not be covered in this thesis.

The primary objective of using ASDMs in organisations is to deliver information systems quickly and to change rapidly and as frequently as possible to the project environment's circumstances (Highsmith, 2002b). Agile system development methodologies focus on incremental and iterative development, and keeping stakeholders and users involved in order to deliver a product that is up to date in a constantly changing project environment. Because of changing environments and requirements, some practitioners in the mid-1990s found documentation requirements and design development steps associated with formal SDMs frustrating, and in some circumstances even impossible (Lindvall *et al.*, 2002:197). Agile system development methodologies were developed to address these problems.

Agile system development methodologies are gaining popularity and many organisations have been adopting ASDMs since their creation in the early 1990s (Good, 2003:27). Agile system development methodologies are relatively new and researchers and practitioners (other than the authors of the ASDMs) do not specifically know the environments and circumstances in which it will function successfully. According to Qumer and Henderson-Sellers (2008a:1899), there are few organisations in practice that can implement a full ASDM in a short period, as full transition normally takes several years. The adoption of ASDMs into business also holds many challenges (Peterson & Wohlin, 2009:1479;³ Chan & Thong, 2009:803) and there are several factors and circumstances that influence the implementation of ASDMs (Livermore, 2008:31; Qumer & Henderson-Sellers, 2008a:1899; Myer, 2008:56). New issues arise when using ASDMs on large-scale projects (Peterson & Wohlin, 2009:1479). In order to overcome the challenge of ASDM acceptance in organisations, Chan and Thong (2009:803) developed a conceptual framework for ASDM acceptance. According to Myer (2008:57), most of the ASDMs have an adaptive nature but they only work within certain circumstances.

Project management was mainly applied by supervisors, engineers and architects until the mid-1900s. The study investigated two structured and widely accepted PMMs – Project Management Body of Knowledge (PMBOK) and Project IN Controlled Environment (PRINCE2) – and a relatively new PMM called Agile Project Management (APM) as explained by Highsmith (2010). PMMs consist of a life cycle, several techniques, tools and processes of managing the project elements (project requirements and expectations, resources, stakeholders, etc.) to satisfy predefined quality, cost, scope and time constraints in order to complete a project successfully.

³ Reference made to a whole document; like full text internet articles, journal articles, newspaper articles, and magazine articles will only contain the first page number.

There is a critical need within organisations for an adaptable and flexible PMM with the ability to deliver projects of different sizes and complexities in a constantly changing environment (PM4DEV, 2008:31; Augustine, 2005:213; Cohn & Schwaber, 2003:10). Currently PMMs and ASDMs struggle to address the problems mentioned above because PMMs are normally too structured and ASDMs present challenges regarding wide acceptance in organisations. This need will be addressed by developing a hybrid agile project management methodology (APMM). The word “hybrid” in this context can be defined as a cross-breed of aspects, characteristics, strengths and best practices from the different PMMs and ASDMs to develop a hybrid APMM with a unique and mixed character. The reason for combining the different PMMs and ASDMs is to ensure that the strengths of each can be utilised to increase the possibility of the hybrid APMM being accepted, implemented and successfully executed in a constantly changing project and business environment. Normally when PMMs and ASDMs are combined in practice, a PMM uses only one ASDM to develop and deliver a product or project.

1.3 Research question

Against the above background, the main research question was posed:

Can a combination of ASDMs and PMMs be used to develop a hybrid APMM that will have the ability to deliver IT projects successfully in a constantly changing business and project environment?

1.4 Research objectives

The research question of this study will be addressed by achieving the following research objectives:

- develop the agile project success criteria;
- develop the ASDM best practice framework;
- develop the PMM best practice framework;
- develop a proposed hybrid APMM (ver. 0) that would address the agile project success criteria;
- develop an improved hybrid APMM (ver. 1) by conducting two case studies; and
- develop an improved hybrid APMM (ver. 2) by conducting a survey.

1.5 Method of investigation

Seven ASDMs and the three different PMMs were studied in order to establish whether a combination of ASDMs and PMMs could be used to develop the proposed hybrid APMM that

would have the ability to deliver IT projects in an environment where change is inevitable. In order to do this, a mixed research paradigm was adopted, combining the positivistic research paradigm (design science and surveying) and the interpretive research paradigm (case study).

A literature review was done to identify and describe the reasons that projects fail and the critical project success factors (CPSFs) required for projects to succeed. The reasons that projects fail and CPSFs will be summarised as the *critical project success criteria*. The factors in the *critical project success criteria* will be categorised according to the different factors of Chow and Chao's (2008:961) five categories (organisational, people, process, technical, project) to create the *agile project success criteria*.

The *ASDM best practice framework* and *PMM best practice framework* were developed by identifying which ASDM or PMM satisfies a specific agile project success factor (APSF) in the *agile project success criteria*. Strengths, weakness, and gaps were summarised across the ASDMs and PMMs for each of the APSFs in the two frameworks. The hybrid APMM (ver. 0) was developed interpretively using design science (research approach) and constructivism by combining the strengths, addressing the weaknesses and bridging the gaps identified in the frameworks. The respective frameworks were thus used as a foundation to develop the proposed hybrid APMM (ver. 0) that would address the *agile project success criteria* to ensure that it has the ability to deliver IT projects successfully in a constantly changing project and business environment.

The hybrid APMM (ver. 0) was evaluated by conducting an interpretive case study. After the qualitative data had been collected and analysed using cross-case analysis, the findings were incorporated to create an improved hybrid APMM (ver. 1). A survey was then conducted by means of questionnaires sent to analysts, programmers, project managers and senior management in order to collect quantitative and qualitative data in evaluating the hybrid APMM (ver. 1). The data collected was statistically analysed and the findings were incorporated into the final hybrid APMM (ver. 2).

1.6 Structure of the thesis

Chapter 2: *Agile system development methodologies*. In this chapter, the terms “system development methodology” (SDM) and “agile system development methodology” (ASDM) will be defined. Then, both the new and relatively well-established seven ASDMs and their effectiveness will be explained. Agile system development methodologies in general will be evaluated by considering their area of application, advantages and disadvantages. Lastly, the seven different ASDMs will be compared using the four elements of an SDM (Huisman & livari,

2006:32) to emphasise some of the main similarities and differences amongst the different ASDMs.

Chapter 3: *Project management methodologies*. In this chapter, the terms “project” and “project management” will be defined. PMBOK and PRINCE2 will then be explained as structured and well-established PMMs in practice, after which a relatively new PMM called Agile Project Management (APM) will be studied. Each PMM will be evaluated to explain its effectiveness and use in practice. Lastly, the three PMMs will be compared.

Chapter 4: *Research design*. In this chapter, the manner in which the research design was applied will be explained. Firstly, the positivistic, interpretivistic, critical social theory and mixed methods paradigms will be explained. Design science as research method will then be explained, after which the research plan will be presented using the design science cycle as foundation. The possibility of merging ASDMs and PMMs will be discussed, by combining positivistic design science, positivistic surveying and interpretive case study research as a mixed methods research approach to creating versions 0, 1 and 2 of the hybrid APMM. The manner in which the different versions of the hybrid APMM were developed will be explained in the research plan, in which the mixed methods paradigm (interpretivism and positivism), research methods (design science, case study and survey), data-collection techniques (interviews and questionnaires), and data-analysis techniques (cross-case and statistical) will be detailed.

Chapter 5: *Development of a hybrid APMM*. In this chapter, the development of the hybrid APMM (ver. 0) will be detailed. The reasons that projects fail and CPSFs will be summarised to form the *critical project success criteria*, which will be used to create the *agile project success criteria*. The different ASDMs and PMMs will be then evaluated using the APSFs in the *agile project success criteria* to develop the ASDM best practice framework and PMM best practice framework, respectively. The findings of each APSF in the respective frameworks were used as foundation to develop the proposed hybrid APMM (ver. 0) interpretively using constructivism.

Chapter 6: *Evaluation of the hybrid APMM*. In this chapter, the testing and evaluation of the proposed hybrid APMM (ver.0 and ver.1) will be reported on by conducting case studies and by conducting a survey. Case studies were conducted at a large (case study A) and small (case study B) organisation. The content of each interview was analysed, after which case studies A and B were compared using content and cross-case analysis. The hybrid APMM (ver. 0) was improved by incorporating the findings of the cross-case analysis to create the hybrid APMM (ver. 1), which was distributed to different respondents across different organisations with a questionnaire in order to collect quantitative and qualitative data. The questionnaires were then

statistically analysed and the findings were incorporated into the hybrid APMM (ver. 1) to develop the final hybrid APMM (ver. 2).

Chapter 7: *Findings and future work*. This chapter will summarise the findings and will explain whether the research question and corresponding research objectives have been addressed. The research objectives will be listed and discussed as stipulated in chapter 1, after which the contribution of the study will be explained by presenting the final hybrid APMM (ver. 2). The limitations of the study will be described, after which the chapter will finish with recommendations for future work.

1.7 Summary

In this chapter, the problem statement and substantiation for this study were explained and the aim of this study was then set out by presenting the research question and objectives. The manner in which these were addressed was described in the method of investigation, after which the chapter division was provided.

The seven relatively well-established ASDMs will be discussed in the next chapter, after an SDM and ASDM have been defined. Each ASDM will be evaluated, after which the effectiveness of ASDMs in general will be explained.

CHAPTER 2

AGILE SYSTEM DEVELOPMENT METHODOLOGIES

2.1 Introduction

In this chapter, the definition of a “system development methodology” (SDM) will be investigated since there is no universally accepted definition. An “agile system development methodology” (ASDM) will then be defined, and what it means for an organisation to be agile will be discussed. Furthermore, the seven core ASDMs commonly used in practice will be described in detail because numerous references will be made to the different aspects of the ASDMs when the development of the hybrid APMM (ver.0) is explained in Chapter 5. These are:

- Dynamic System Development Methodology (DSDM);
- Scrum;
- Extreme Programming (XP);
- Feature Driven Development (FDD);
- Crystal ASDMs;
- Adaptive Software Development (ASD); and
- Lean Development (LD).

The effectiveness of each individual ASDM at implementation level will be discussed with reference to papers and surveys conducted by experts in agile project development. Each individual ASDM will be evaluated according to the agile characteristic features and values as provided by Qumer and Henderson-Sellers (2008b:280).

Agile system development methodologies in general will be evaluated by considering their area of application, advantages, and disadvantages. Lastly, a comparison will be done between the seven different ASDMs using the four elements of an SDM (Huisman & livari, 2006:32) to emphasise some of the main similarities and differences amongst the different ASDMs.

2.2 Definition of “system development methodology”

Trying to define an SDM is a difficult task. There is no universally accepted, concise definition of information SDM (Avison & Fitzgerald, 2003:527; Wynekoop & Russo, 1997:48; livari, Hirscheim & Klein, 1999:1). The first difficulty in trying to define an SDM is the method versus methodology debate. Researchers have different views. Some argue that the term “methodology” has no place in information systems, because it literally means a “science of methods” (Schach, 1997:23), while others argue that the terms can be applied interchangeably

(Hardy, Thompson & Edwards, 1995:467–468; Saeki, 1998:925). Others argue that methodologies encompass methods or that methods encompass methodologies (Palvia & Nosek, 1993:73).

There are conceptual difficulties related to the use of the term “system development method/methodology”. livari and Maansaari (1998:502–503) classify these difficulties as “scope problems” and “category problems”. Scope problems include instances in which a system development method/methodology covers the system’s development process, or in which there is concern about the aspects that the system development method/methodology should cover. Category problems include difficulty in distinguishing between techniques and system development methods/methodologies.

Despite category problems and scope problems, four elements can be identified in the various definitions of a system development method/methodology (Huisman & livari, 2006:32):

- the system development method/methodology itself;
- a system development method/methodology is based on some philosophical view or approach;
- a system development method/methodology includes a set of techniques;
- a system development method/methodology follows a process model.

Avison and Fitzgerald (2003:20, 527) argue that the term “methodology” is a much wider concept than the term “method” because a methodology has certain characteristics that are not implied by method, and a methodology includes a philosophical view. In this study, the researcher uses the term “methodology” to cover all four elements, because it is a much larger concept than the term “method”, and does not aim to contribute to the method versus methodology debate. Therefore, the term “system development methodology” will be used, instead of “system development method”.

Consequently, an SDM can be defined as a combination of the following (Huisman & livari, 2006:32):

- *A system development approach*: This is the philosophical view on which a methodology is based. It is the set of goals, fundamental concepts, guiding principles and beliefs of the system development process that drive interpretations and actions in system development (livari, Hirscheim & Klein, 1998:165–166; livari *et al.*, 1999:2). Examples of system development approaches include the process-oriented approach, object-oriented approach, information modelling and user-centric approach.
- *A system development process model*: Wynekoop and Russo (1993:182) define a “process

model” as a representation of the sequences of stages through which a system evolves. Examples of process models are the waterfall model, linear life cycle, spiral model, and incremental and iterative model.

- *A system development method*: According to Brinkkemper (1996:275), this is an approach to developing a system, based on a line of thought that consists of certain rules and definitions, prearranged systematically into activities related to products to be delivered. He also states that it is a “way of investigation”. Wynekoop and Russo (1993:182) describe a method as a systematic approach to conducting at least one complete phase of system development, consisting of a set of guidelines, activities, techniques and tools and based on a particular philosophy of system development and the target system. Examples include Information Engineering, Structured Systems Analysis and Design Method, Jackson Systems Development and XP.
- *A system development technique*: This technique can be described as a procedure, possibly with a prescribed notation, to perform a development activity (Brinkkemper, 1996:276). Examples include entity relationship diagrams, decision tables, data-flow diagrams, and pair programming.

Understanding the elements of an SDM is essential to understanding the ASDM. The term “agile system development methodology” will be defined in the following section.

2.3 The agile system development methodology

The ASDM will be discussed by first considering the Agile Software Development Manifesto’s twelve guiding principles, after which an ASDM will be defined. The five characteristic features and six characteristic values will be explained which will be used to measure the agility of each ASDM in Section 2.4. The guiding principles and values of an ASDM will also be summarised.

2.3.1 Agile Software Development Manifesto

Several agile leaders were called together to create the Agile Software Development Manifesto in 2001. The following twelve guiding principles were agreed upon that would govern all ASDMs in seeking to be flexible and more successful (Beck *et al.*, 2001):

- Principle 1: “Our highest priority is to satisfy the client through early and continuous delivery of valuable software.”
- Principle 2: “Welcome changing requirements, even late in development. Agile processes harness change for the client’s competitive advantage.”
- Principle 3: “Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.”
- Principle 4: “Business people and developers must work together daily throughout the

project.”

- Principle 5: “Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.”
- Principle 6: “The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.”
- Principle 7: “Working software is the primary measure of progress.”
- Principle 8: “Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.”
- Principle 9: “Continuous attention to technical excellence and good design enhances agility.”
- Principle 10: “Simplicity – the art of maximizing the amount of work not done--is essential.”
- Principle 11: “The best architectures, requirements, and designs emerge from self-organizing teams.”
- Principle 12: “At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.”

The reason for presenting these twelve guiding principles is that they will be used to better understand Agile Project Management (APM) as explained in Section 3.5.

2.3.2 Definition of “agile system development methodology”

The primary goal of any ASDM is to make an organisation agile, in other words, give the organisation the ability to adapt to change. However, the following question arises: what does it mean to be agile? Highsmith (2002b) states that it means being able to deliver quickly, change quickly, and to change as often as necessary. Highsmith and Cockburn (2001:120) explain that what is new about ASDMs is not the practices they use, but their recognition of people as the primary drivers of project success, coupled with a primary focus on manoeuvrability, change and effectiveness. Fowler (2005) calls ASDMs “new methodologies” because, owing to their adaptive nature and people-first orientation, they have blossomed during the past ten years. Agile system development methodologies come from rapid development, rapid prototyping and a resurgence of identifying programming not as an industrial process but as a craft (Boehm & Turner, 2003:16). To define an ASDM effectively, the following attributes associated with ASDMs must be understood (Lindvall *et al.*, 2002:201; Boehm & Turner, 2003:16):

- *Iterative*: The system is delivered incrementally where each increment is developed iteratively. An increment is released after its functionality has been changed because of new requirements.
- *Incremental*: The system is delivered in pieces. This means that the requirements are partitioned into small subsystems, after which the new requirements and functionality are

added.

- *Self-organising*: The team is obliged to manage and organise themselves in order to complete the system within budget and time constraints.
- *Emergent*: Technology and requirements are allowed to emerge throughout the product development cycle. Agile system development methodologies have the ability to adapt to change so that new requirements can emerge and be implemented.

There is no universal accepted definition of ASDM (Larman, 2004:25), but Boehm and Turner (2003:17) define ASDMs as “very lightweight processes that employ short iterative cycles; actively involve users to establish, prioritise, and verify requirements; and rely on tacit knowledge within a team as opposed to documentation”. Qumer and Henderson-Sellers (2008b:281) define an ASDM as a methodology that is “people focused, communications-oriented, flexible (ready to adapt to expected or unexpected change at any time), speedy (encourages rapid and iterative development of the product in small releases), lean (focuses on shortening time-frame and cost and on improved quality), responsive (reacts appropriately to expected and unexpected changes), and learning (focuses on improvement during and after product development)”.

Conboy and Fitzgerald (2004:108) explain that agility consists of two components, namely flexibility and speed. Terms such as “fast”, “rapid”, “speed” and “quick” are commonly found in definitions of “agility”; thus, for an organisation to be agile it must be able to respond “speedily and flexibly”. An ASDM is more “adaptive than predictive”, more “people-oriented than process-oriented” (Fowler, 2001). Other agile concepts include the use of time-boxes, inspections and continual delivery (Pence, 2007:36–37).

Based on the definitions and descriptions of ASDMs presented above, an ASDM can be summarised in the following definition, which will be used during this study:

An ASDM is an adaptable/flexible SDM that focuses on integrated communication, user involvement, minimised documentation, and incremental and iterative development to deliver quickly and as often as possible in a constantly changing environment.

2.3.3 Characteristics of an agile system development methodology

According to Boehm and Turner (2005:30), ASDMs are characterised by the ability to handle changing business requirements, incremental and iterative development, and continuous code integration. Qumer and Henderson-Sellers (2008b:282) characterised ASDMs according to a set of five ASDM features and six ASDM values:

Five ASDM characteristic features:

- flexibility;
- speed;
- leanness;
- learning; and
- responsiveness.

Six ASDM characteristic values:

- individuals and interactions valued above processes and tools;
- working software valued above comprehensive documentation;
- client collaboration valued above contract negotiation;
- responding to change valued above following a plan;
- keeping the process agile; and
- keeping the process cost effective.

These five characteristic features and six characteristic values form part of the 4-Dimensional Analytical Tool (4-DAT) framework developed by Qumer and Henderson-Sellers (2008b:280). The 4-DAT framework examines ASDMs in relation to four dimensions and is part of the Agile Software Solution Framework (ASSF), which will be explained in Section 2.5. The agility of each ASDM, except LD (it was not investigated by Qumer & Henderson-Sellers, 2008b:280) will be explained using these five characteristic features and six characteristic values in the following section.

2.4 The seven agile system development methodologies

In this section, only the seven core ASDMs used in practice will be explained. These ASDMs were developed in sequence, starting with DSDM in 1994 and ending with the most recently popularised ASDM, LD, as presented in Figure 2.1.

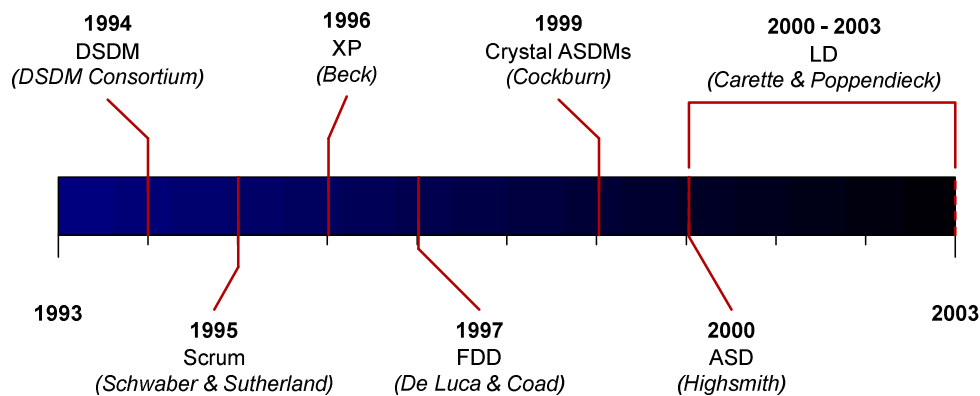


Figure 2.1: A historical timeline of the development of the ASDMs

According to Highsmith (2002a:6), these ASDMs are the core group as set out in the Agile Software Development Manifesto (Beck *et al.*, 2001; Highsmith, 2002a:6; Lindstrom & Jeffries, 2004:44). This group is viewed as the early initial ASDMs (Lindstrom & Jeffries, 2004:44). Furthermore, the core set of ASDMs is relatively well established in practice according to Schuh (2004:17).

2.4.1 Dynamic System Development Methodology

The Dynamic System Development Methodology (DSDM) was defined in January 1994 when the sixteen founding members of the DSDM Consortium met for the first time (Hislop, Lutz, Naveda, McCracken, Mead, & Williams 2002:176; DSDM Consortium, 2005; Stapleton, 1997:xv–xvi). Their goal was to jointly develop and promote an independent Rapid Application Development (RAD) framework and to expand the proven successes of RAD by providing a “framework of controls for rapid and responsive delivery”. A high-level framework was produced that was approved unanimously by the 36 members.

The Dynamic System Development Methodology is not so much a methodology as it is a framework, and the basic concepts have remained the same although the framework has been refined over time (DSDM Consortium, 2005). According to this source, DSDM “has been found to be applicable in nearly every technical and business environment where systems are needed quickly”. According to Stapleton (1997:xiv), DSDM is not about tools, but people and quickly delivering cost-effective, workable solutions that satisfy business requirements. The design of DSDM is based on certain principles, the first five of which define the principles that guided DSDM’s structure, while the last four explain the foundation on which DSDM is built (Stapleton, 1997:xvi; Highsmith, 2002c:xxxii; Daughtrey, 2002:48; Baltus, 2001:20–24):

- A cooperative and collaborative approach amongst all stakeholders is important.
- Testing must be integrated through the DSDM life cycle.

- Business and user requirements are base-lined at a high level.
- Changes during the development process can be reversed.
- In order to create an accurate solution, incremental and iterative development is required.
- The critical criterion for deliverable acceptance is fitness for business purpose.
- Frequent delivery of products is the major focus.
- The team must have the authorisation and mandate to make decisions.
- Active user involvement is essential.

Abrahamsson, Salo, Ronkainen and Warsta (2002:63) state that the main idea behind DSDM is to keep time and resources fixed while adjusting functionality accordingly, keeping business and user requirements in mind (see Figure 2.2). The fundamental idea of DSDM differs from the traditional approach, where the extent of functionality of a product is fixed, and time and resources must be adjusted to reach this fixed level of functionality. While functionality is allowed to vary, time boxes are used to maintain control. In this way, systems can be implemented speedily and without difficulty. In this environment, these systems can serve as the basis for further evolution. The Dynamic System Development Methodology is an extension to RAD practices and it boasts the best-supported documentation and training of any other ASDM (Highsmith, 2002a:8).

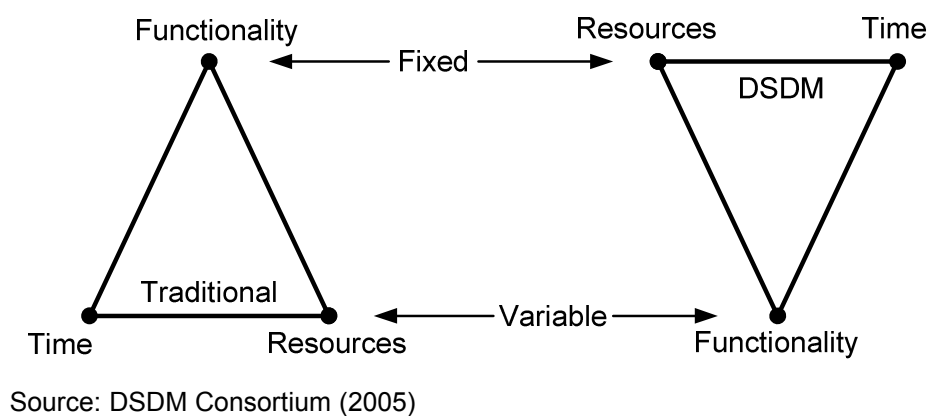
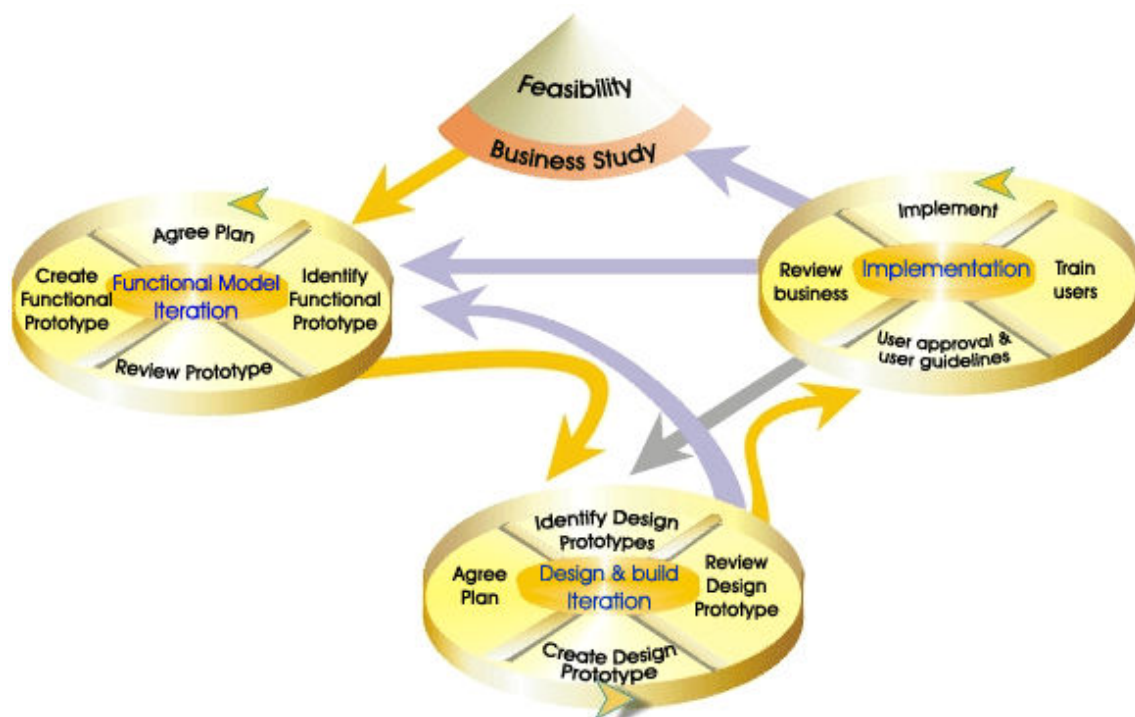


Figure 2.2: The traditional approach versus the DSDM approach

The philosophy of the DSDM framework that guides the thought of DSDM developers (DSDM Consortium, 2005) is as follows. Firstly, development can be incremental, which means that the whole development process can be defined in increments (pieces). Each increment can then be designed, developed, tested and deployed. Secondly, development is viewed as a team effort, in which the knowledge of IT professionals and clients are combined. Thirdly, the available resources must initially be spent to develop the most important business requirements. Lastly, to deliver a product of high quality, the organisation must be technically advanced and quick to meet demands.

The Dynamic System Development Methodology is “lightweight”, meaning that it does not focus on documentation. One of the principles of this methodology is the importance of collaboration, that is, the use of prototypes to capture information rather than bulky documentation (Highsmith, 2002a:8). The Dynamic System Development Methodology offers a more complete, defined development process like most well known ASDMs.

The Dynamic System Development Methodology identifies five distinct phases: feasibility study, business study, functional model iteration, design and build iteration, and implementation. These are preceded by the pre-project phase and concluded with the post-project phase, as shown in Figure 2.3. It is up to the individual project to decide on the manner in which to work through the phases (Schuh, 2004:39), as it is an incremental and iterative approach.



Source: Stapleton (1997:3)

Figure 2.3: The DSDM life cycle

Pre-project: This phase ensures that everything is in place and set up (Doom, 2010:71) correctly, that funding is available, and that the project is ready to begin successfully (Zelkowitz, 2004:20). A brief assessment is undertaken to determine whether DSDM can be utilised to execute the project (Daughtrey, 2002:48).

The feasibility and business study: Both these studies are time-boxed and done sequentially. The business study usually takes a month to complete, while the feasibility study usually takes several weeks. The business case investigates the business and technical requirements and constraints, as well as the fundamental components of the project

(Daughtrey, 2002:48; Doom, 2010:71). The primary goal of the feasibility study is to determine whether DSDM is the correct approach for a specific project (Zelkowitz, 2004:20; Hislop *et al.*, 2002:176; Cohen, Lindvall & Costa, 2003:19).

In evaluating the type of project, with people and organisational issues as primary concerns, a decision should be made whether DSDM should be used to manage the project (Abrahamsson *et al.*, 2002:64). The feasibility study is also concerned with the technical and technological possibilities of developing the project, and the risks that may be involved.

In the business study phase, the essential business and technological characteristics are analysed and prioritised (Abrahamsson *et al.*, 2002:65). This phase entails working together, using facilitated workshops attended by “empowered and knowledgeable staff who can quickly pool their knowledge and gain consensus as to the priorities of the development” (Cohen *et al.*, 2003:19).

As a result, the business area is defined, describing the affected business processes with their information needs, markets and identified users. Early client identification ensures early client involvement. Other outputs in the business study phase are the systems architecture definition, which is the “first system architecture sketch” and has the ability to change as the project develops, and the outline prototyping plan, which describes the “prototyping strategy for the following stages, and plan for configuration management” (Abrahamsson *et al.*, 2002:65).

If DSDM is deemed appropriate for the proposed project, the business study scopes the overall activity and sets the framework for both technical and business activities (Hislop *et al.*, 2002:176). After these two phases, the high-level requirements are base-lined, system architecture is outlined and the functional and information models are produced (Hislop *et al.*, 2002:177).

Functional model integration: A number of functions are identified for implementation by building a repeatable functional prototype, which is repeated to include all functionality in the application (Doom, 2010:71). The primary aim is to build each prototype according to the high level of processing and information requirements explained and identified in the business study phase (Zelkowitz, 2004:20; DSDM Consortium, 2005). This phase is the first “iterative and incremental phase” (Abrahamsson *et al.*, 2002:65). At each iteration, the approach is planned, reviewed and then analysed in order to be applicable in subsequent iterations. The experience gained through coding, analysis and prototype building is used to improve the analysis model. The functional model is produced using the analysis models and prototype code.

The functional model provides four outputs (Abrahamsson et al., 2002:65):

- *prioritised functions*: the prioritised list of functions delivered at the end of each iteration;
- *functional prototyping review documents*: comments from users about the current increment that can be used for other increments;
- *non-functional requirements*: requirements that should be met during the next phase; and
- *risk analysis for further development*: important document for the functional model iteration phase, as problems will be more difficult to correct from the next phase onwards.

Cohen *et al.* (2003:20) state that this phase and the design and build phase have a common process:

- identify what is to be produced;
- agree on the manner in which and by when to do it;
- create the product;
- check whether it has been produced correctly.

Design and build iteration: The prototypes from the functional model iteration are completed, combined and tested to create a system of sufficient internal and external quality to be safely released to the users (Zelkowitz, 2004:22; Cohen *et al.*, 2003:20). The output of the design and build phase is a tested system that meets at least the most important requirements set by users. Like the *functional model iteration* phase, this phase is repeated until all functional and non-functional requirements are satisfied (Doom, 2010:72). After the users have reviewed the design and functional prototypes, all further development is based on the users' comments and requirements (Abrahamsson *et al.*, 2002:66). Testing is not a distinct phase, but very important and integrated into the DSDM life cycle (Hislop *et al.*, 2002:177).

Implementation: In this phase, the system is implemented within the operational user organisation and responsibility for operation is transferred to the users after completion of the necessary tests (Daughtrey, 2002:48; Hislop *et al.*, 2002:177). An increment review document is created during this phase in which the state of the system is reported on. At this stage, the system can be either complete, meeting all requirements, or incomplete where some functionalities may be missing, or only some requirements (not even primary requirements) are met. If the system has not been completed, the functional model iteration, design and build iteration, and implementation phases are repeated until the system has been fully completed (Cohen *et al.*, 2003; Doom, 2010:72; Zelkowitz, 2004:22). If the implementation is done over a period, this phase may also be iterated (Abrahamsson *et al.*, 2002:66). The output of the implementation phase is a user manual, explaining the manner in which to gain maximum usage of the system, and a project review document, which summarises the outcome of the

project and explains the reason for potential further development based on the project results.

Abrahamsson *et al.* (2002:66) define four possible courses of development for DSDM. Firstly, if the system meets all requirements, no further work needs to be done. Secondly, if some requirements were not met because they were only discovered during the development stage, the process may be repeated. Thirdly, if some less-critical function has to be omitted, the process may be repeated from the functional model iteration phase. Lastly, if some technical issues had to be ignored owing to time constraints, they may be addressed by iterating again, starting from the design and build iteration phase.

The key is delivering what the business needs when it needs it. This is done by using the various techniques in the framework and flexible requirements. The aim is always to address the current and imminent needs of the business rather than to attack the perceived possibilities (DSDM Consortium, 2005).

Post-project: The main concern is the maintenance of the implemented system and clean-up, during which the remaining issues are resolved and the last actions are completed (Doom, 2010:72; Zelkowitz, 2004:20). According to the DSDM Consortium (2005), maintenance is done by keeping the project solution operating effectively.

Effectiveness and evaluation of the Dynamic System Development Methodology according to agile system development methodology characteristics

The Dynamic System Development Methodology has been tried and tested, was found to be very stable and effective, and practitioners view it as the “de facto RAD standard in the UK” (Aydin, Harmsen, Van Slooten & Stegwee, 2004:130), which is spreading to Africa, Benelux, Denmark, Europe, France, India, North America and Sweden (Baltus, 2001:19). The Dynamic System Development Methodology has been applied to small and large projects, and used in combination with other ASDMs (DSDM Consortium, 2005). The Dynamic System Development Methodology is an ASDM that “strongly emphasises the concepts of suitability and adaptability” at project and organisation level, and it can be adapted if it is not entirely suitable (Aydin *et al.*, 2004:130). The disadvantage of DSDM is that it does not “specifically address team size, exact iteration lengths, distribution, or system criticality” (Zelkowitz, 2004:20) unlike the Crystal ASDMs.

The Dynamic System Development Methodology does not support leanness (see ASDM characteristic features in Section 2.3, as explained by Qumer & Henderson-Sellers, 2008b:282), which results in DSDM not supporting the sixth ASDM characteristic value of keeping the

process cost effective. It does however support the other five ASDM characteristic values.

DSDM presents more project management aspects than other ASDMs by using project management related phases such as the *pre-project*, *implementation* and *post-project* phases. This makes DSDM a potentially effective combination with other PMMs. The combination between DSDM and PRINCE2 was found to be very successful as explained in Section 3.4.4. If DSDM can be effectively combined with elements of PRINCE2, it can surely also adopt strengths of other PMMs.

2.4.2 Scrum

Ken Schwaber and Jeff Sutherland developed Scrum in 1995. It was first described in 1996 by Cohen *et al.* (2003:13) as a process that accepts the unpredictability of the development process with a “do what it takes mentality”, and it has been implemented successfully by numerous independent software vendors.

Scrum is a dynamic and iterative ASDM that releases product functionality regularly. Scrum is based on the following: three roles – product owner, Scrum master and team; three documents and processes – product backlog, sprint backlog and sprint results; and three meetings – sprint or pre-sprint planning, daily scrum, and post-sprint or sprint review meeting (Stober & Hansmann, 2010:41–42).

Scrum is a well-established and well-documented ASDM (Stamelos & Sfetsos, 2007:14). The name “Scrum” is borrowed from the game of rugby. A scrum takes place when the eight players of each team, called the forwards, bundle together, and push and shuffle against the opponents for possession of the ball. In order to win the ball, the one team must displace the other from its current location.

The primary idea of Scrum is that system development involves requirements, resources, technology, and time constraints, which are likely to change during development. This changing environment makes the development process very complex and unpredictable. Thus, the system development process requires flexibility and adaptability to respond suitably to the changes during development (Abrahamsson *et al.*, 2002:27). Scrum is an “empirical approach applying the ideas of industrial process control theory to system development resulting in an approach that reintroduces the ideas of flexibility, adaptability and productivity” (Abrahamsson *et al.*, 2002:27). It focuses on producing a system that is flexible by using a team to produce such a system within a constantly changing environment.

Scrum is primarily concerned with a few key management tasks rather than the manner in

which the product is actually developed (Good, 2003:18). IT projects are divided into iterations called “sprints” that take thirty days or less to complete, through which a set of features is delivered. The management of the project’s progress takes place according to the method called “scrum”, also known in some cases as “brainstorming”, which entails a daily meeting held for fifteen minutes by the management team (Good, 2003:20).

Scrum is explained according to the Scrum life cycle (see Figure 2.4). The product backlog (see Figure 2.4) contains the body of work required during the entire project. This includes requirements gained from software developers, clients and experts. All requirements are prioritised according to descending order of importance. Owing to the dynamic environment, the product backlog must be constantly updated and prioritised as new requirements are identified. The product owner is defined as the person responsible for the product backlog and he/she represents the expectations, constraints, and interests of the stakeholders (Kammermeier, 2009:5; Haugan, 2011:289). He/she also ensures that funding is provided to the project and signs off on any deliverables completed (Stober & Hansmann, 2010:41). The product backlog is part of planning, which “includes the definition of the system being developed, project team, tools and other resources, risk assessment and controlling issues, training needs and verification management approval” (Abrahamsson *et al.*, 2002:29). The architecture phase is completed together with planning, which includes the changes and those problems the changes may cause in implementing the product backlog if the implemented system requires enhancement (Abrahamsson *et al.*, 2002:29). A design review meeting is held to implement these changes.

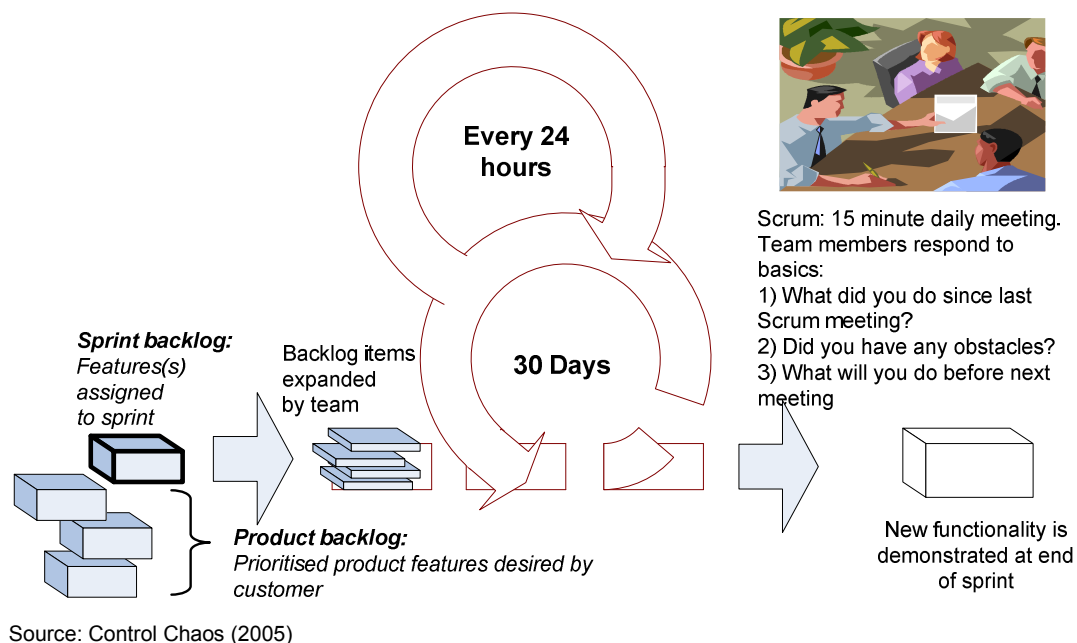


Figure 2.4: The Scrum life cycle

A “sprint” is a period of up to thirty days during which sectioned tasks will be performed to create deliverables that satisfy the requirements set by users, managers and experts (Huijbers *et al.*, 2004:17). The self-organising project team is the primary actor responsible for completing deliverables of quality (Kammermeier, 2009:5). Because development is incremental, each sprint includes the traditional phases of software development namely requirements, analysis, design, evaluation, documentation and delivery (Abrahamsson *et al.*, 2002:30; Stober & Hansmann, 2010:42). There can be as many as eight sprints when Scrum is used to develop a system.

Before a sprint is undertaken, the team has to undertake pre-sprint planning, which entails identifying the tasks necessary to reach the defined sprint goal. These identified requirements and tasks are moved from product backlog to sprint backlog to be completed during the next sprint after they have been prioritised (Cohen *et al.*, 2003:14; Zelkowitz, 2004:15). After the first sprint, a post-sprint meeting is held during which a decision is made regarding whether the team should continue with the project. If this is agreed upon, a pre-sprint meeting is held to identify tasks to be completed during the next sprint.

The sprint backlog is the starting point for every sprint, which contains all the tasks and requirements that ought to be completed during the current sprint (Cohen *et al.*, 2003:14). The tasks to be performed during the current sprint are selected by the Scrum team, the Scrum master and product owner during pre-sprint planning (also known as the sprint planning meeting), using the prioritised list of the product backlog (Abrahamsson *et al.*, 2002:33). The Scrum master is responsible for the overall project’s success, including the delivery of products of good quality, facilitation of communication, removal of impediments, and the process as a whole (Kammermeier, 2009:5; Stober & Hansmann, 2010:41; Haugan, 2011:289). This includes adhering to the rules, values, process and practices of Scrum.

Every morning, a short meeting of approximately fifteen minutes is held to keep track of the development process. During each meeting, the Scrum team specifies what has been done since the last meeting, and discusses what should be done before the next meeting takes place (Abrahamsson *et al.*, 2002:34). During these meetings, problems are identified and solutions are suggested to keep the team focused on the goal. Furthermore, the meeting enhances communication by keeping the stakeholders and team members involved and up to date (Zelkowitz, 2004:15). These meetings can also take the form of short and powerful stand-up meetings, at which definite problems can be discussed and quick solutions found. This helps the team to believe in their own abilities and helps the client believe in the team (Abrahamsson *et al.*, 2002:35).

After each sprint, a post-sprint meeting or sprint review meeting (Abrahamsson *et al.*, 2002:34; Stober & Hansmann, 2010:41) is held amongst the Scrum master, team and product owner to analyse the progress and to demonstrate the system to management and clients (Zelkowitz, 2004:15). This meeting may result in new requirements to improve the system. These new requirements are added to the prioritised product backlog and sprint backlog. The next sprint is then planned, based on the prioritised product backlog and sprint backlog.

Cohen *et al.* (2003:14) summarise the key principles of Scrum:

- “[s]mall working teams that maximize communication, minimize overhead, and maximize sharing of tacit, informal knowledge”;
- “[a]daptability to technical or marketplace (user/client) changes to ensure the best possible product is produced”;
- “[f]requent builds, or construction of executables, that can be inspected, adjusted, tested, documented, and built on”;
- “[p]artitioning of work and team assignments into clean, low coupling partitions, or packets”;
- “[c]onstant testing and documentation of a product as it is built”; and
- “[a]bility to declare a product done whenever required”.

According to Schwaber and Beedle (2002:59), Scrum can be adopted in existing projects and new projects. For a new IT project, a product backlog must first be built by working with the team and clients for several days. The first sprint will then involve key pieces in system development. These key pieces include an initial system framework, technological requirements and business functionality. The sprint will entail tasks regarding setting up the team roles, building management practices and fulfilling the sprint goal. As the Scrum team members work with the sprint backlog, the product owner works with the clients to build a more comprehensive product backlog. This will enable them to plan the next sprint after the first post-sprint meeting.

Effectiveness and evaluation of Scrum according to agile system development methodology characteristics

Scrum is an ASDM that can be used to manage both large and small projects, although it was designed to work with small teams of ten or fewer people (Livermore, 2008:32). Different types of IT projects ranging from financial products to Internet solutions have been developed in more than fifty different organisations using Scrum. Scrum is utilised worldwide by hundreds of organisations such as Google, Microsoft, Motorola, SAP, Nokia Siemens Networks (Cooke, 2010:123). Many organisations have adopted and approved of Scrum, for example Microsoft who “welcomes the use of methodologies like Scrum” (Taft, 2005:15). Some organisations,

such as HP and Intel, only make partial use of Scrum for project planning and tracking because of their global software development organisational environment (Holmstrom, Fitzgerald, Agerfalk & Conchuir, 2006:13).

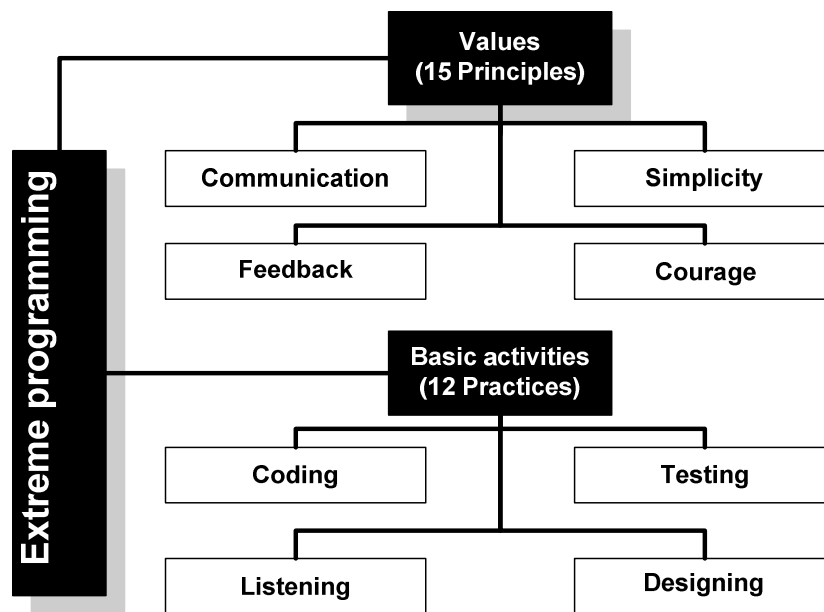
Scrum grants the opportunity to make fact-based judgments for satisfying immediate client expectations (Stober & Hansmann, 2010:44). Furthermore, Scrum assists other processes within an organisation to become agile (Hunt, 2006:26). Scrum can be introduced at the start of a project or during a product's life cycle – which makes it very adaptable when building a hybrid approach (Hunt, 2006:26). Scrum can be adopted in an environment in which a project with its own technology already exists and in cases in which teams are struggling to cope with increasingly newer technology and changing requirements. During the first sprint, user functionality should be demonstrated on the existing system technology (Schwaber & Beedle, 2002:59).

Like DSDM, Scrum supports the first five ASDM characteristic values, except the sixth value of keeping the process cost effective. In addition, Scrum does not support leanness (an ASDM characteristic feature; Qumer & Henderson-Sellers, 2008b:284–285).

Scrum offers a very logical and easy manner in which to break work to be completed down. Because work is broken down into logical sprints, it is a very effective way of managing the development of complex systems, while a normal SDM would be overwhelmed by the amount and complexity of work to be completed. Owing to its ease of use, no complications should arise when combined with other PMMs. It would also be easy to extract specific components from Scrum to be combined with other methodologies because of its logical structure and distinctive life-cycle components.

2.4.3 Extreme Programming

Extreme Programming (XP) was first introduced in 1996 by Kent Beck while serving as project leader on Chrysler Comprehensive Compensation – a long-term project to rewrite Chrysler Corp's payroll application. It was further popularised by Beck (2000). Numerous articles published subsequently further popularised XP. It is by far the most popular ASDM to emerge in recent years (Highsmith, 2002a:7; Bowers *et al.*, 2007:283; Cohen *et al.*, 2003:12; Lindstrom & Jeffries, 2004:43) and it is "often the first ASDM that companies intend to implement" (Tolfo & Wazlawick, 2008:1955). Extreme Programming owes much of its popularity to developers' disenchantment with traditional methods that do not work in certain environments. Developers started looking for something new, something extreme, and something that would work in a constantly changing environment.



Source: Hislop *et al.* (2002:173)

Figure 2.5: The XP structure

Extreme Programming can best be explained in terms of its structural components, as shown in Figure 2.5 above. The four values, *communication*, *simplicity*, *feedback* and *courage*, yield *fifteen principles* that are unique to XP project development. The four basic activities, *coding*, *testing*, *listening* and *designing*, can be viewed as XP's backbone, which forms the basis of the *twelve core practices*.

The four values of Extreme Programming

According to Lindstrom and Jeffries (2004:50), the values of XP can be used to test whether the ASDM suits the project, team and organisation. The team's actions are guided by these values. Lindstrom and Jeffries (2004:45) state that XP is the only ASDM that is "explicit in its values" – that yields principles – and its practices, as seen in Figure 2.5. This explicit combination gives guidance on the manner in which to react if the practices do not work (using the values), and what to do (using the practices; Lindstrom & Jeffries, 2004:45).

Communication: Extreme Programming is considered "lightweight" because it focuses exclusively on communication amongst team members, as well as between the team and its clients. The communication should be of high quality for both team members and clients. Development is guided by clients communicating functional requirements as "stories" written on small file cards (Hislop *et al.*, 2002:172), named "story cards". Software-engineering studies have found that the majority of project disasters were caused by a breakdown of communication between developers and clients, internally amongst the developers, and internally amongst the

clients (Holcombe & Holcombe 2008:21–22), which is the reason that communication plays such an important role. Teams and stakeholders that regularly communicate face-to-face in an honest and open manner solve problems quicker than those who do not do so (Pearman & Goodwill, 2006:5).

Simplicity: This requires that systems developers and designers build the simplest system (Burke & Coyner, 2003:11) that will satisfy the requirements set by sponsors and business users. As the requirements are implemented in the evolving system, system designers and developers must be cautious not to make the design too complex while implementing the necessary modifications. The team must only build what is required (Chromatic, 2003:8) to satisfy the identified requirements, instead of including functionality in the system that will never be used (Pearman & Goodwill, 2006:5). System modification entails the improvement of code structure while preserving system functionality (Hislop *et al.*, 2002:173).

Feedback: Constant and continuous involvement of clients, system designers and developers result in immediate and rapid feedback on the status and progress of the system being developed to ensure that the project team stays on the right track (Pearman & Goodwill, 2006:5; Hislop *et al.*, 2002:173; Holcombe & Holcombe 2008:22; Chromatic, 2003:8). Feedback plays a large role in XP, for example clients define their requirements on story cards, and developers provide valuable feedback to clients on the progress made to satisfy the clients' requirements. Continuous testing provides programmers with rapid feedback on errors within the design code. Pair programming (which will be explained later in the section) also provides a continuous feedback loop (Hislop *et al.*, 2002:174).

Courage: The developers are responsible for developing a system that is simple, user-friendly, and that fulfils most requirements. Creating the simplest design may require courageous decisions, such as throwing out large chunks of code or re-engineering the system to eliminate duplicate code (Hislop *et al.*, 2002:174). The team must have the necessary courage to handle the resistance to change they might encounter with regard to the work they are doing (Pearman & Goodwill, 2006:6). This means making tough decisions when required (Chromatic, 2003:10), admitting mistakes, and fixing them.

Holcombe and Holcombe (2008:25) mention a fifth value, namely *respect*. Many of the problems encountered during the execution of a project are “people issues”, which are normally associated with relationships. In order to manage relational and people issues, the team members and all other stakeholders involved must treat each individual with the necessary respect they deserve – by allowing them to explain their perspective, by resolving conflict in an

orderly manner, and by maintaining a healthy working relationship.

The fifteen principles of Extreme Programming

There are fifteen principles that are built on the XP values and that bridge the gap between the four XP values and the practices that implement the values (Hislop *et al.*, 2002:173; Pearman & Goodwill, 2006:6):

- assuming simplicity;
- incremental changes;
- embracing change;
- high-quality work delivery;
- teach and learn XP;
- small initial investments;
- playing to win;
- concrete experimentation;
- accepting responsibility;
- promoting honest and open communication;
- working with people's instincts;
- adapting as required;
- travelling light;
- honest measurement; and
- rapid feedback.

The four basic activities of Extreme Programming (Baird, 2002:29–30)

Coding: In XP, coding is seen as a learning activity. Extreme Programming is designed for programming, which is a profession that is enhanced by XP practices such as pair programming and refactoring (Baird, 2002:29). “Coding helps the developer to test his/her thinking process, because if the thinking process is correct the code will do what it is designed to do.” (Hislop *et al.*, 2002:174).

Testing: Testing is a continuous process throughout system development (Baird, 2002:29), as new requirements are incorporated into the evolving system. Developers must listen to clients in order to know which requirements should be tested (Hislop *et al.*, 2002:174). Testing every completed task ensures that the system design is correct and that all tasks are up to date.

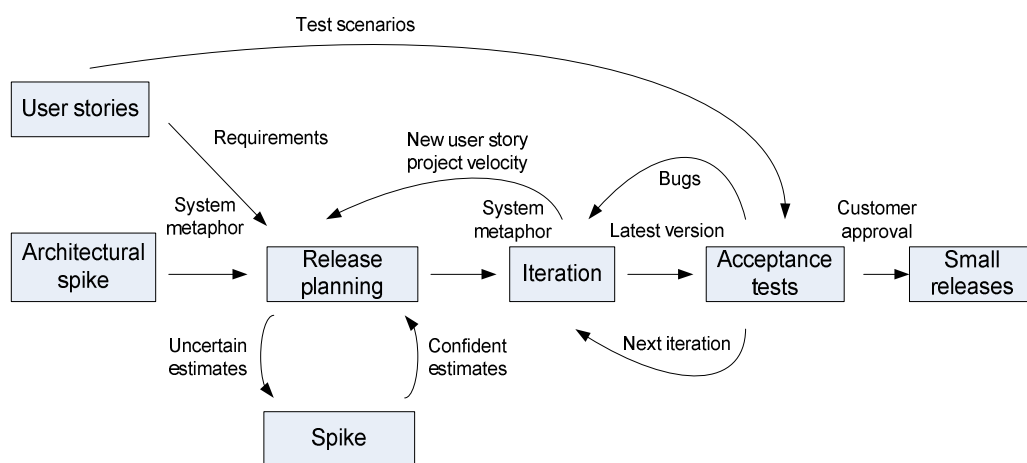
Listening: If system designers and developers do not know how to listen to users, they will not know how to design the system. Extreme Programming simplifies listening by the use of story

cards. It is important for developers to listen to clients' stories so they can help refine these in order to know what should be tested and developed (Hislop *et al.*, 2002:174). It relies on valuable verbal communication and active listening because of its reduced dependence on formal documentation (Baird, 2002:29).

Designing: In XP, designing is a continuous activity of incorporating new tasks and requirements into the evolving and already existing system. Extreme Programming is fluid and has a responsive sense of design, as it views design as dynamic and team based not stationary (Baird, 2002:30). It relies on a metaphor to guide its design (Hislop *et al.*, 2002:174). A metaphor is a brief description that conveys the system's main function attributes (this will be explained in the twelve core practices of XP).

The twelve core practices of Extreme Programming

The twelve core practices of XP are explained in this section using the XP life cycle. The XP life cycle and the manner in which the twelve practices of XP are integrated into it are demonstrated in Figure 2.6. In the figure, a *spike* can be explained as an “exploring effort by developing prototype code”, which has a specific goal in mind – it is not general technical research (Baird, 2002:59).



Source: Adapted from Wells (2000)

Figure 2.6: The XP life cycle

- **The planning game:** As each iteration starts, managers, clients, and developers meet to flesh out, prioritise and predict what should be accomplished before the next release, and estimate the requirements for the next release (Cohen *et al.*, 2003:12; Lindstrom & Jeffries, 2004:47). The two key planning steps are release planning (the client presents the desired requirements to system programmers, and the programmers estimate their difficulty) and iteration planning (the team is given direction every few weeks as the system develops;

Lindstrom & Jeffries, 2004:47). A release is broken into iterations of one to three weeks each. The requirements are called “user stories” and are captured on story cards in a language all parties will understand (Cohen *et al.*, 2003:12). The time it will take and the cost to develop the necessary functionality that would satisfy the user stories are estimated (Holcombe & Holcombe 2008:28; Pearman & Goodwill, 2006:12; Crispin & House, 2003:6). In this practice, the user lists the requirements of the system.

- **Small releases:** Extreme Programming teams practise releases in two ways (Lindstrom & Jeffries, 2004:47). Firstly, at each iteration, the team produces tested, running software that is of value to clients. Secondly, the team releases software of business value to their end users as frequently as possible. Development is incremental, which means that a basic system is implemented quickly, where new releases are implemented at least every month until the required (whole) system is up and running (Abrahamsson *et al.*, 2002:23; Good, 2003:23). Projects developed in shorter time-frames have found to be more successful than those that have lengthy delivery dates (Pearman & Goodwill, 2006:13), as the team stays motivated because of being able to see what they have accomplished to date.
- **Metaphor:** Managers, clients and programmers define a metaphor or a set of metaphors that guides all development by describing the manner in which the system works (Abrahamsson *et al.*, 2002:24; Cohen *et al.*, 2003:12; Lindstrom & Jeffries, 2004:49). The purpose is to have a complete and simple way to explain what must be completed by using a defined set of metaphors (Pearman & Goodwill, 2006:13). A collection of classes (grouped programme code) is organised to satisfy the requirements on the story cards that are being developed (Holcombe & Holcombe 2008:29).
- **Testing (client tests and test-driven development):** There are two types of tests that are carried out. Firstly, unit or functional tests are conducted, through which programmers ensure that the code written does what they expect it will do. In this way, programmers gain immediate feedback on their progress (Lindstrom & Jeffries, 2004:48). Secondly, acceptance tests are conducted by developers – but “driven by the client” (Burke & Coyner, 2003:21). This involves written acceptance tests for coding the application to ensure that the system does what the user expects it to do (Cohen *et al.*, 2003:12). These tests are normally done upfront (Pearman & Goodwill, 2006:11) and created loosely based on the current requirements that are available to the project team (Holcombe & Holcombe 2008:26). Tests are not conducted only at the end of the project, but form an integral part of development and are thus conducted continually as the project progresses (Baird, 2002:34).
- **Simple design:** Developers are urged to start simple and keep it simple, although new requirements are identified and modifications should be made during development (Lindstrom & Jeffries, 2004:48; Crispin & House, 2003:5). The system should be designed

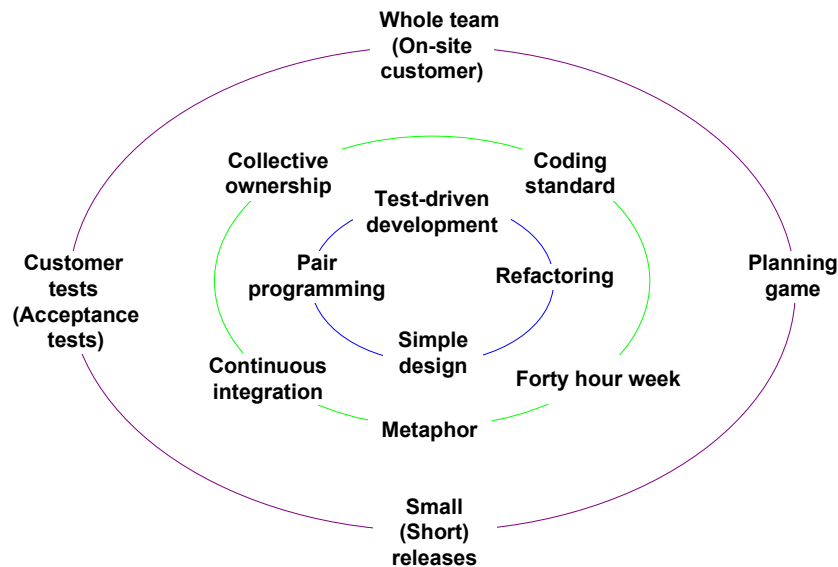
and implemented as simply as possible. The reason for using simple design is to create a system that is easy to use, maintain and manage. According to Baird (2002:34), “simple systems are easy to extend, work faster, are easier to communicate about, and simpler to maintain”.

- **Refactoring:** When a design is no longer appropriate or of value, it should be changed. Refactoring involves improving the design and code by keeping the system functional and simple through removing duplication, improving communication and adding flexibility (Abrahamsson *et al.*, 2002:24; Burke & Coyner, 2003:22; Lindstrom & Jeffries, 2004:48). This results in a more understandable and maintainable solution (Holcombe & Holcombe 2008:33).
- **Pair programming:** This involves two programmers programming in front of the same PC, where both are intimately involved in writing and reviewing the code (Crispin & House, 2003:5; Abrahamsson *et al.*, 2002:24; Lindstrom & Jeffries, 2004:48; Burke & Coyner, 2003:15). This practice ensures effective coding with fewer errors in a shorter time (Pearman & Goodwill, 2006:10; Holcombe & Holcombe 2008:26; Chong *et al.*, 2005:43), while the one programmer learns from the other. This practice is also effective if one programmer falls ill because the other can continue without time being lost. While one programmer programmes, the other “thinks more strategically” about where the approach will work, and about ways to simplify the design (Avison & Fitzgerald, 2003:444). Reasons for doing something a certain way are open for discussion (Holcombe & Holcombe 2008:26), which on the other hand may cause time being wasted on non-relevant debates as each person has his/her own way of doing things.
- **Collective ownership:** Every developer has ownership of all development documents and program code, and can make modifications anywhere and at any time, while system development takes place to remove duplication (Pearman & Goodwill, 2006:10; Crispin & House, 2003:6; Abrahamsson *et al.*, 2002:24; Cohen *et al.*, 2003:12; Holcombe & Holcombe 2008:32). The benefits of collective ownership are increased code quality and reduced defects, as well as elimination of code duplication by different programmers (Lindstrom & Jeffries, 2004:49).
- **Continuous integration:** Developers integrate a new piece of coding as soon and as often as possible (Abrahamsson *et al.*, 2002:24; Cohen *et al.*, 2003:12; Crispin & House, 2003:5) to ensure that the team members are equally up to date at all times (Burke & Coyner, 2003:27). Every time a task is completed, the completed task is integrated into the system, after which tests are run, which must be passed in order for the new changes in the code to be accepted (Abrahamsson *et al.*, 2002:24; Cohen *et al.*, 2003:12; Lindstrom & Jeffries, 2004:48). This will result in quickly resolved integration issues (Pearman & Goodwill, 2006:12).

- **Forty-hour week:** Developing software is very demanding. Therefore, as a rule XP states that team members work no more than 40 hours per week. This will ensure that the “development teams do not take on more work than they have proven to be capable of completing” (Pearman & Goodwill, 2006:13). Overtime is allowed, but not for more than two subsequent weeks (Abrahamsson *et al.*, 2002:24; Cohen *et al.*, 2003:12). This prevents programmers and developers burning out or making errors that may harm the project. In many organisations, this practice will not be implemented.
- **On-site client:** Using XP, communication is enriched by allowing the client on-site at all times as part of the team, contributing to development, answering questions, performing acceptance tests, and ensuring that the development progresses as expected (Abrahamsson *et al.*, 2002:24; Good, 2003:23; Cohen *et al.*, 2003:12; Holcombe & Holcombe 2008:27; Pearman & Goodwill, 2006:9; Crispin & House, 2003:4). The on-site client ensures that the developers stay focused on the requirements. If they lose focus, the client is there to help the developers regain focus in order to satisfy the project requirements.
- **Coding standards (open work-space):** The programmers of an XP project follow a common coding standard so that all code that emphasises communication appears to have been written by one individual (Pearman & Goodwill, 2006:10; Crispin & House, 2003:5; Abrahamsson *et al.*, 2002:24; Cohen *et al.*, 2003:12; Holcombe & Holcombe 2008:32).

Pearman and Goodwill (2006:14) added two additional practices to these twelve practices, namely *sit together* and *journalising*. *Sit together* is an implied XP practice that refers to the team sitting together in the same work-space. *Journalising* refers to writing down what happened during the day (thoughts, observations, etc.) to identify certain patterns and lessons learnt which can be used to enhance communication or streamline development.

Extreme Programming practices can be grouped into three different cycles (see Figure 2.7). The outer cycle reflects the practices that affect all the project participants, the middle cycle relates to the work of the development team, and the innermost cycle relates to the work of the developers (Lindstrom & Jeffries, 2004:46).



Source: Adapted from Jeffries (2001); Lindstrom and Jeffries (2004:46)

Figure 2.7: The practices and main cycles of XP

The rhythm of an Extreme Programming project

According to Lindstrom and Jeffries (2004:45), an XP project has a rhythm that proceeds in iterations of two weeks. During each iteration, a set of requirements is developed and tested. Figure 2.8 shows the activities of the programming team and clients during the iterations of an XP project. As the project steadily progresses, the client chooses when the entire project (with maximum functionality) is delivered (Lindstrom & Jeffries, 2004:45).

	Week 0 Preparing for first iteration	Week 1 and 2 iteration 1	Week 3 and 4 iteration 2	Week n the rhythm continues..
Customer team	Exploration Story writing Alignment Customer tests	Prepare for next iteration Manage release schedule Communicate with end users/stakeholders Customer tests	Prepare for next iteration Manage release schedule Communicate with end users/stakeholders Customer tests	
Programming team	Agree on process exploration	Daily stand-up Sit together Discuss detailed requirements Run customer tests continuously	Daily stand up Sit together Discuss detailed requirements Run customer tests continuously	
	Spikes Development Environment Preparations	Tasks Continuous builds Some spikes Estimation of new stories for later iterations	Tasks Continuous builds Some spikes Estimation of new stories for later iterations	

Source: Lindstrom and Jeffries (2004:46)

Figure 2.8: The rhythm of an XP project

Effectiveness and evaluation of Extreme Programming according to agile system development methodology characteristics

Lindstrom and Jeffries (2004:50) state that the most commonly debated question regarding XP is whether it can be applied successfully to a particular type (different environments) of project. The extent of use, application and adoption of XP differ from project to project and organisation to organisation (Salo & Abrahamsson, 2008:63). Like Scrum, XP can be used to manage both small and large projects (Livermore, 2008:32). Intel and HP only use parts of XP in their organisational environment for technical engineering when global software development is done (Holmstrom *et al.*, 2006:13).

Experience demonstrates that system development using XP is effected and limited by the culture of the organisation, the organisational environment, characteristics of the project, the people on the team, and effective XP practice adoption (Lindstrom & Jeffries, 2004:50; Tolfo & Wazlawick, 2008:1955; Bowers *et al.*, 2007). The business contexts in which XP is applied might threaten its success. For example, lawyers, accountants and executives might wish to see detailed and formalised financial or contractual documentation put in place before approval is provided (Holcombe & Holcombe, 2008:35). Extreme Programming is still evolving and the lessons learnt are being incorporated as the ASDM matures.

Extreme Programming does not support leanness (ASDM characteristic feature) and the two most recently identified agile characteristic values of keeping the process agile and keeping the process cost effective (Qumer & Henderson-Sellers, 2008b:283-284). In order to evaluate whether the XP practices can help a team achieve greater success, consideration must be given to these limitations.

Extreme Programming may become too complex owing to its four values, yielding *fifteen principles*, and four basic activities, which form the basis of the *twelve core practices* that integrate with the XP lifecycle. Although it might become too complex, it does however offer the possibility of easy combination with other methodologies, as it should be easy to extract specific values, principles, activities and practices from XP because of its distinctive structure.

2.4.4 Feature Driven Development

Feature Driven Development (FDD) was created by Jeff De Luca and Peter Coad in 1997 and later published in De Luca and Coad (1999). This ASDM was created when De Luca and Coad were brought in as consultants on a project in trouble (the large lending system project at United Overseas Bank in Singapore). Coad applied feature-oriented development techniques to the project, while De Luca used a streamlined, lightweight process framework (Hislop *et al.*,

2002:175). They then merged their concepts into an ASDM that became known as FDD to save the highly complex Singapore project from failure (Hislop *et al.*, 2002:175; Cohen *et al.*, 2003:17). The previous developers spent two years on the same project, writing over 3 500 pages of documentation without any code (Highsmith, 2002a:5) and declared the project undoable (Hislop *et al.*, 2002:175; Cohen *et al.*, 2003:17). Feature Driven Development was applied to the failing project and began delivering a product in increments to a surprised client within two months (Hislop *et al.*, 2002:175; Highsmith, 2002a:5).

Feature Driven Development relies on a basic architecture that is represented as Unified Modelling Language (UML) class diagrams, which are developed early in a project. The FDD ASDM primarily focuses on the design and building phases of the software development process (Abrahamsson *et al.*, 2002:47). It is built according to a core set of “best practices” that complement and reinforce each other (Palmer & Felsing, 2002:35). The creators of FDD do not maintain that they have bought a new SDM to market – they used a management technique to integrate proven industry best practices (Anderson, 2004:181; Haugan, 2011:288).

Best practices in FDD include (Palmer & Felsing, 2002:35; Westfall, 2009:139–140):

- *Domain object modelling*: The problem domain is explained and explored to deliver a framework to which features can be added.
- *Developing by feature*: The progress is tracked via a list of small, functionally decomposed and client-valued functions.
- *Individual class (code) ownership*: The responsibility of performance, consistency and conceptual integrity of a class are assigned to an individual.
- *Feature teams*: These are small, dynamic teams.
- *Inspection*: This is done using the best-known defect-detection mechanisms.
- *Regular builds*: These ensure that there is always a running basic system available to which new features can be added.
- *Configuration management*: The latest versions of each source code file are identified and tracked through configuration management.
- *Progress reporting*: This involves reporting on all completed sections at organisational level.

The lead designers decompose these business practices into features to plan, design and code these features (Abrahamsson *et al.*, 2002:47). Features are small items useful for users, just like story cards used in XP, written in an understandable language for all parties that should not take longer than two days (Cohen *et al.*, 2003:18). Feature Driven Development encapsulates best practices and incremental development to manage and monitor the development process and, when completed, deliver features to clients in two-week cycles (Abrahamsson *et al.*,

2002:47; Hislop *et al.*, 2002:175).

Cohen *et al.* (quoted in Highsmith, 2002c:273) explain some core values that would work best for developing a project using FDD. These values are:

- A way for building IT systems is necessary in order to scale to larger projects.
- A simple, well-defined process works best.
- Process steps should be logical, and their value immediately obvious to each team member.
- “Process pride” (developers believing that their process will work, although there is a process that will work much better in the current situation) can keep the real work from happening.
- Good processes move to the background, so team members can focus on results.
- Short, iterative, feature-driven life cycles are best.

The FDD ASDM consists of five sequential processes, entailing techniques, guidelines, roles, goals, artefacts, timelines and methods that can be used in FDD project development. Feature Driven Development is portrayed using the ETVX (Entry criteria, Task, Verification, eXit) pattern (Highsmith, 2002c:275; Schuh, 2004:28). The five processes are illustrated in Figure 2.9.

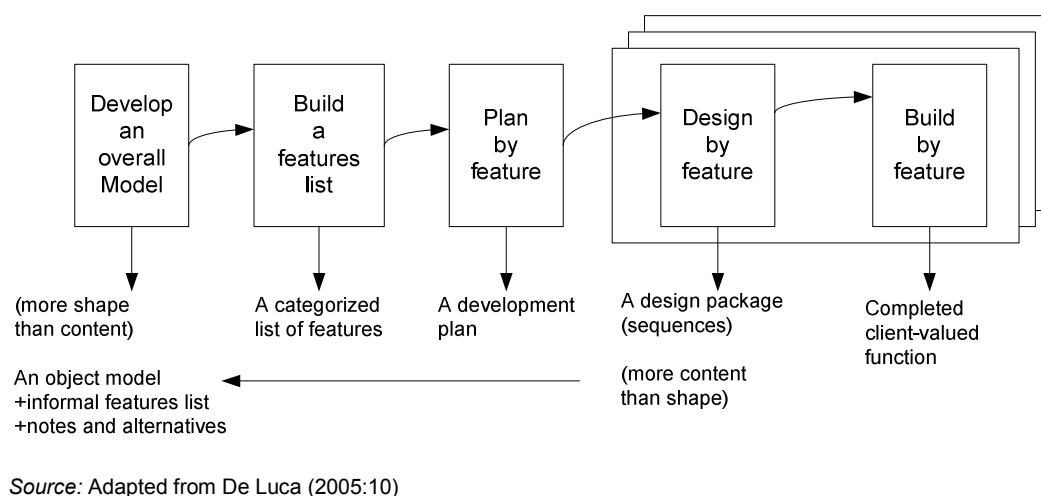


Figure 2.9: FDD processes

Develop an overall model: In the figure above, the FDD process begins with a high-level walkthrough with the client (Haugan, 2011:289) and “the domain experts become familiar with the scope, context and requirements of the system” (Westfall, 2009:138). These domain experts present a walkthrough version of the project, in which the chief architect, who is responsible for the overall system design, and members of the team are informed of the primary requirements and system description (Abrahamsson *et al.*, 2002:48, 52; Westfall, 2009:138).

The overall domain is divided into various domain areas. A detailed walkthrough demonstrates that every domain is concerned with producing object models for each domain area (Abrahamsson *et al.*, 2002:48). An overall model is constructed by selecting the appropriate object models for each domain. This process executes the following tasks (Highsmith, 2002c:275):

- The modelling team is formed.
- A domain walkthrough is conducted.
- Documents are studied.
- The model is developed.
- The overall project model is refined.
- Model notes are written.
- External and internal assessments are conducted.

This process ensures that the project team has a clear understanding of what the system must consist of and it ensures that any misunderstandings are resolved in the early stages of the project (Schuh, 2004: 29).

Build a feature list: During the next step the information from the walkthroughs and the identified requirements are used by the team to create a list of client-valued features or functions of the system (Highsmith, 2002c:275; Westfall, 2009:138; Schuh, 2004:29; Haugan, 2011:289). These functions (or function groups) consist of major feature sets that represent each of the domain areas. The major feature sets are further divided into feature sub-sets that “represent different activities within specific domain areas” (Abrahamsson *et al.*, 2002:49; see Figure 2.10). Features that will take longer than ten days to implement are divided into sub-features (Cohen *et al.*, 2003:18). The sponsors and users review the feature list, after division, to determine whether the list is complete and relevant.

Plan by feature: The next step is developing a high-level plan of prioritised features and their dependencies (Westfall, 2009:139) after which the feature sets are assigned to chief programmers (Cohen *et al.*, 2003:18; Westfall, 2009:138). Chief programmers are experienced developers who lead small teams in the analysis, design and development of new features. Chief programmers also assign class ownership and responsibility to other individual developers. In this process, the estimated time for feature sets and features are estimated (Rychly & Ticha, 2007:200)

Design by feature and build by feature: These processes are focused on the design, build and implementation of individual features by the feature owner (Rychly & Ticha, 2007:200). This iterative process begins by the chief programmer selecting a small group of

features from the feature set(s) (see Figure 2.10) “by combining features that use the same classes” (Westfall, 2009:139). The selected features are planned in more detail, built, tested and integrated iteratively within two weeks (Abrahamsson *et al.*, 2002:49; Cohen *et al.*, 2003:18; Schuh, 2004:29).

After the successful completion of each iteration, which includes coding, code inspection, testing and integration (Westfall, 2009:139), the current iteration is released and the next iteration is started. During this iteration, the chief programmer chooses new features from the feature set(s). The composition of the feature list, feature sets and features is demonstrated in Figure 2.10.

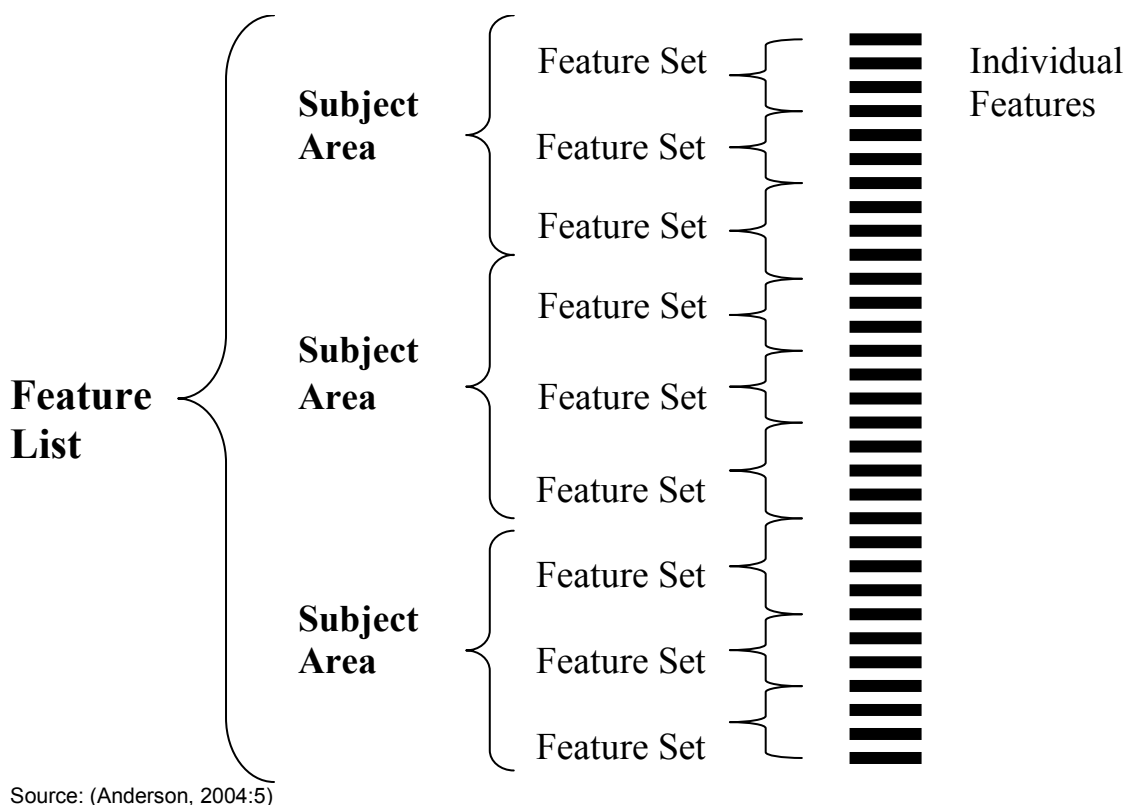


Figure 2.10: Component assembly in FDD

Figure 2.10 explains the component assembly of FDD very simply. A feature list consists of several subject areas. Each subject area consists of several feature sets. Finally, a feature set consists of individual features.

Effectiveness and evaluation of Feature Driven Development according to agile system development methodology characteristics

The FDD ASDM promises early and frequent incremental delivery of working code if

documentation is minimised. Unlike XP, FDD needs an architectural design in the form of class diagrams and analysis of features using sequence diagrams (Hislop *et al.*, 2002:176). Similar to other ASDMs, this ASDM promises early and continuous client involvement throughout the project (Hislop *et al.*, 2002:176). Feature Driven Development has the ability to incorporate other ASDM's techniques (Livermore, 2008:32). Large projects or teams that work in an environment in which upfront design is viewed as important can benefit by integrating FDD into their current development processes (Schuh, 2004:29).

Feature Driven Development has a high success rate with large projects if there is diverse talent within the project, that is, competent domain experts, developers and chief programmers (Highsmith, 2002a:6). It is also suitable for new projects, projects in need of code upgrading, and projects that require the development of a second version (Abrahamsson *et al.*, 2002:54). It would be wise to adapt this ASDM in small increments as development progresses (Abrahamsson *et al.*, 2002:54). According to Rychly and Ticha (2007:179), the application of FDD "leads to better project management and consistency of a software's design, implementation and documentation".

Like DSDM and Scrum, FDD supports the first five agile characteristic values, except the sixth value of keeping the process cost effective (Qumer & Henderson-Sellers, 2008b:285) because it does not support leanness (an ASDM characteristic feature).

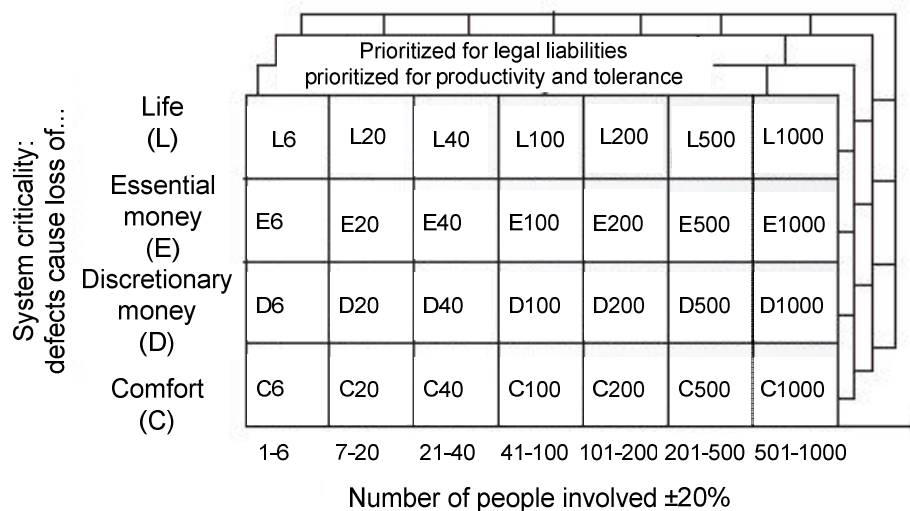
Feature Driven Development offers a structured approach and thus does not support leanness. Although it breaks work down into manageable segments, it offers no adequate change management, an absent component that would make this ASDM even more effective. This concern could be addressed by merging FDD with PMMs that adequately address change management in order to improve FDD's ability to deliver IT projects in a constantly changing project and business environment.

2.4.5 Crystal agile system development methodologies

Crystal ASDMs were created by Alistair Cockburn in 1999 with the aid of interviews with real-world practitioners. He aims to address different kinds of project requirements with different kinds of Crystal ASDMs. Crystal ASDMs are "people focused, communication-centric, and highly tolerant" (Schuh, 2004:30) with the ability to adapt to different types and sizes of IT projects. The Crystal ASDM in use must be continually reviewed and tailored as the project progresses because of its adaptive nature (Rosenberg, Stephens & Collins-Cope, 2005:18). Crystal ASDMs focus on people, the interaction between people and the community, but primarily on face-to-face communication. This philosophy is explained by Highsmith and

Cockburn (2001:120) as follows: if one replaces written documentation with face-to-face interaction, one could improve the likelihood of delivering a system in frequent running pieces and reduce its reliance on documentation. Crystal project development is incremental with a maximum increment length of four months, but preferably between one and three months (Abrahamsson *et al.*, 2002:37).

Crystal ASDMs consist of a family of ASDMs, which gives developers the option to choose the most appropriate methodology for a specific project. This family consists of twenty ASDMs and can be presented on a two-dimensional grid (Rico, Sayani & Sone, 2009:31), as shown in Figure 2.11 on the next page. In the figure, the Y-axis represents the criticality of the system or the project that must be completed, while the X-axis represents the number of people involved in a project team.



Source: Highsmith (2002c:264)

Figure 2.11: The framework of Crystal ASDMs

In the figure, the symbols C, D, E and L are an indication of potential loss caused by a system failure (Abrahamsson *et al.*, 2002:36). The critical level symbols stand for: comfort (C), discretionary (D), essential money (E), and life (L). Abrahamsson *et al.* (2002:36) explain that the criticality level C indicates a system crash because of defects that cause a loss in comfort for users, whereas L indicates a defect in the critical system that may literally mean a loss of life. Every block within the graphic (X- and Y-axes) of Figure 2.11 represent a project category symbol. For example, D6 is a project with a maximum of six people delivering a system of maximum criticality of D (Abrahamsson *et al.*, 2002:37).

Each Crystal ASDM has a colour that describes its difficulty; the darker the colour, the more difficult the ASDM. There are seven main colours: clear, yellow, orange, red, maroon, blue, and

violet. According to Abrahamsson *et al.* (2002:36), Crystal ASDMs suggest choosing the most applicable colour for a project based on its criticality and size. In a large team (501–1 000 people), the system criticality of life is prioritised as the most difficult according to Figure 2.11. According to Cohen *et al.* (2003:16), Crystal ASDMs can also be adapted to other priorities such as productivity or legal liability. By adding people to the project, one moves to the right on the grid to a darker version of a Crystal ASDM. As the project's criticality increases, the methodology becomes more difficult, and one moves upwards on the Y-axis.

The main factors contributing to the colour selection of Crystal ASDMs are:

- volume of communication required amongst the members of the team;
- occurrence of threatening implications if problematic software is realised; and
- involvement of corporate entities or subjects that might complicate the process of developing and delivering the project.

The most commonly used Crystal ASDM is Crystal Clear (CC), followed by Crystal Yellow, Crystal Orange (CO), Crystal Red, etc. (Cohen *et al.*, 2003:16). The Crystal ASDM used depends on the degree of importance of communication and the number of people involved. Crystal ASDMs do not specify the technical or organisational environment and culture in which they function (Qumer & Henderson-Sellers; 2008b:288). This known set of Crystal ASDMs expands as the ASDM becomes more difficult or the project team expands (Cohen *et al.*, 2003:16).

Crystal Clear

Crystal Clear is designed for a very small project (category D6 projects) with a maximum team member count of six people in one team (Schuh, 2004:32), sharing office space because of the importance of communication. However, Abrahamsson *et al.* (2002:38) explain that if communication (face-to-face) and testing are extended, CC can also be applied to E8/D10 projects. According to Cockburn (2005:307), "CC is a highly optimized way to use a small, collaborate team prioritising for safety in delivering a satisfactory outcome, efficiency in development, and habitability of the working conventions."

Crystal Clear is part of the family of Crystal ASDMs in which every ASDM is identified and characterised by a colour as stated earlier. The more people the team consists of, the darker the colour becomes and the more difficult the project becomes. Every Crystal Clear project has to have seven properties. The first three properties are mandatory, while the last four will allow the project to succeed. However, all seven are desired in a project.

Seven properties of Crystal Clear:

1. *Frequent delivery*: Crystal Clear is an ASDM (explained in Section 2.3), which means that development is done incrementally and iteratively. Work completed is tested every few months (periods should be no longer than two months), and working code must be delivered and implemented into the system. This results in continuous client involvement; consequently, there is constant feedback on the implemented requirements, as well as satisfied sponsors and clients.
2. *Reflective improvement*: Users identify flaws in the system, while iterative development and implementation take place. Requirements that have not been met are identified, and the project team is given time to improve on deficiencies.
3. *Osmotic communication*: Crystal Clear focuses on face-to-face communication. It would be wise to keep the team working closely together, if possible in the same room, so that all questions and problems can be answered and solved. This will result in a neutral work environment, in which team members can acquire relative information, just like osmosis.
4. *Personal safety*: Team members and users must feel safe and comfortable in the project environment they work in. This will result in team members having the confidence to offer constructive criticism on the work of other team members, and to assume responsibility for their own mistakes. This will lead to trust amongst team members because they are honest with one another.
5. *Focus*: There should not be any distractions that can cause team members to lose concentration, such as long meetings and other activities that require multitasking. This will mean that team members will remain focused on their primary objectives and work will be completed much quicker.
6. *Easy access to expert users*: Questions associated with quality and design could be answered by experts available who assist the team during system development.
7. *Technical environment with automated tests, configuration management and frequent integration*: It is critical to have a proper technical environment in which testing and controlling tasks, for example making backups and merging changes, do not have to be done manually in order to make life easier for developers. This will result in the project being completed in less time.

Crystal Clear consists of only one team working in the same office, to whom the following roles are assigned (Cockburn, 2000:17):

- project sponsor;
- user(s);
- senior programmer-designer; and
- programmer-designer.

The most important team member is the senior programmer-designer, who has the reasonability of delivering products such as use and test cases, sequence of releases, design notes and sketches, screen shots, object model(s), code, and a user manual for training (Cockburn, 2000:17).

The seven policy standards of CC are the same for CO, except that CO increments are two to four months, while CC increments are two months or less. The mandatory policy standards for CC and CO are (Abrahamsson et al., 2002:39; Cockburn, 2000:16):

- incremental delivery on a regular basis (CC – two months or less; CO – two to four months);
- progress tracked by milestones based on software deliveries and major decisions rather than written documentation;
- direct user involvement;
- automated regression testing of functionality;
- at least two users should review each release;
- downstream activities start as soon as upstream is “stable enough to review”; and
- workshop held for product and methodology tuning at the beginning and halfway through each increment.

Crystal Orange

Crystal Orange is a Crystal ASDM project that falls within the D40 project category, which consists of ten to forty team members who work in the same building on a system that may cause discretionary monies loss (Cockburn, 2000:15; Williams, 2007:218). It is similar in nature to CC, but the delivery cycle duration can be extended to four months (Schuh, 2004:34). Crystal Orange is viewed as a medium-sized project with the following characteristics (Cockburn, 2000:15):

- ten to forty team members in total;
- one- to two-year project duration;
- time to market is crucial;
- need to communicate with current and newly appointed staff, and there is a need to keep time and cost to a minimum; and
- the system is not life critical.

During the development of CO, Cockburn kept the number of deliverables to a minimum, which reduced their maintenance costs, but included a sufficient number of deliverables to ensure continuous communication amongst team members. He furthermore tailored teams and job

assignments to make the project adjustable to its environment. Crystal Orange is a “common sort of project, requiring trade-offs between complete, extensive deliverables and rapid change in requirements design” (Cockburn, 2000:15).

Staff members that are involved in a CO project can participate in different teams, which are responsible for external testing, infrastructure, functions, technology, architecture, project monitoring, and system planning. The roles of project team members within these above-mentioned teams are typically project sponsor, business expert, business analyst, project manager, different kinds of architects, designer, programmer, technician and tester (Cockburn, 2000:16).

The work products are developed in such a manner that other team members can review them. The work products developed during a CO project include a project schedule, release plan, requirements and development documentation, object model(s), code, and test cases (Cockburn, 2000:16).

Effectiveness and evaluation of Crystal agile system development methodologies according to agile system development methodology characteristics

Crystal Clear has a restricted communication structure and is only suitable for a single team working in one office space. According to Abrahamsson *et al.* (2002:46), CC lacks system validation elements and thus it cannot be applied to life-critical projects. Crystal Clear values the properties of frequent delivery, reflective improvement, osmotic communication, personal safety, focus, easy access to expert users, technical environment with automated tests, configuration management and frequent integration above techniques (Huijbers *et al.*, 2004:21). This means that team members use their own techniques to satisfy the seven CC properties. Therefore, there is not a specific list of techniques that need to be used in order to ensure CC's success.

Uniquely, Crystal ASDMs are the only ASDMs that support all six agile characteristic values (Qumer & Henderson-Sellers, 2008b:288). At a process level, CC does not support leanness (an ASDM characteristic feature), but it does support leanness at a practice level, which makes CC distinct from the other ASDMs evaluated by Qumer and Henderson-Sellers (2008b:280).

Crystal ASDMs are the only ASDMs that truly adapt to different-sized projects and are thus the only ASDM that supports leanness. The ability of Crystal ASDMs to adapt to different-sized projects of various complexities offers an advantage when combined with any other methodologies. The Crystal ASDMs framework (see Figure 2.11) could be utilised by a project when it is initiated in order to determine the project's complexity, which is a component that the

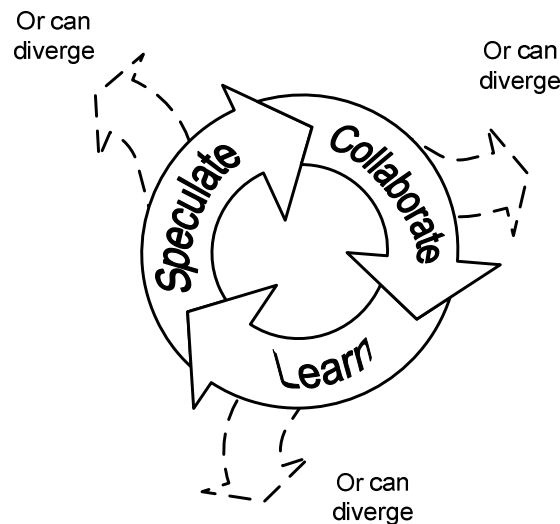
other ASDMs do not provide.

2.4.6 Adaptive Software Development

Adaptive Software Development (ASD) was developed by Jim Highsmith and first documented in Highsmith (2000a). It focuses on rapid, incremental and iterative software development with constant prototyping to develop complex, large systems (Abrahamsson *et al.*, 2002:68; Qumer & Henderson-Sellers, 2008b:285). Adaptive Software Development takes into consideration that the system requirements are not complete at the beginning of the project and that they can change as the project progresses (Lenz & Moeller, 2004:44). A project executed using ASD is “mission focused, feature based, iterative, time-boxed, risk driven and change tolerant” (Schuh, 2004:35). It also focuses on results, not tasks, which are identified as application components.

According to Highsmith (2000b:23), ASD is designed for extreme projects in which high speed, frequent change and a high level of uncertainty are typical. This ASDM emphasises change, and change is positive because it prepares clients and developers for the future. There are many projects that are not extreme, but for those who are complicated and extreme, ASD works better than traditional software development approaches (Highsmith, 2000b:23).

The key notion of ASD is the replacement of the traditional project management *plan-build-revise* life cycle with an adaptive *speculate-collaborate-learn* life cycle (Ambler & Constantine, 2002:261; see Figure 2.12). According to the traditional approach, uncertainty in the planning phase is viewed as a weakness that can evolve into failure. Highsmith (2000b:23) states: “The new ASD lifecycle is dedicated to continuous learning that is geared to constant change, re-evaluation, peering into an uncertain future, and intense collaboration amongst clients, developers and testers.” The phases in the ASD life cycle are named in a way to emphasise the role of change in the development process (Highsmith, 2000b:23).



Source: Highsmith (2000b:23)

Figure 2.12: The ASD life cycle

Highsmith (2000b:24) defines “speculation” as the recognition of the uncertain nature of complex problems and encourages experimentation and exploration. As with APM, the word “planning” is avoided, as ASD predicts that the requirements will change as the project progresses (Lenz & Moeller, 2004:44). This does not mean that planning should not be done (Schuh, 2004:35), it just emphasises that an individual cannot plan uncertainty away. It gives developers and designers room to explore, and allows them to realise that they are unsure, which will allow them to change their plans without fear. Speculation means keeping delivery cycles short and encouraging iteration; it merely acknowledges the reality of uncertainty and it does not mean that planning is abandoned. Deviations must not be viewed as mistakes but as learning opportunities. “Speculating allows us to admit that we don’t know everything – and once we admit to ourselves that we are fallible, then learning becomes more likely.” (Highsmith, 2000b:24).

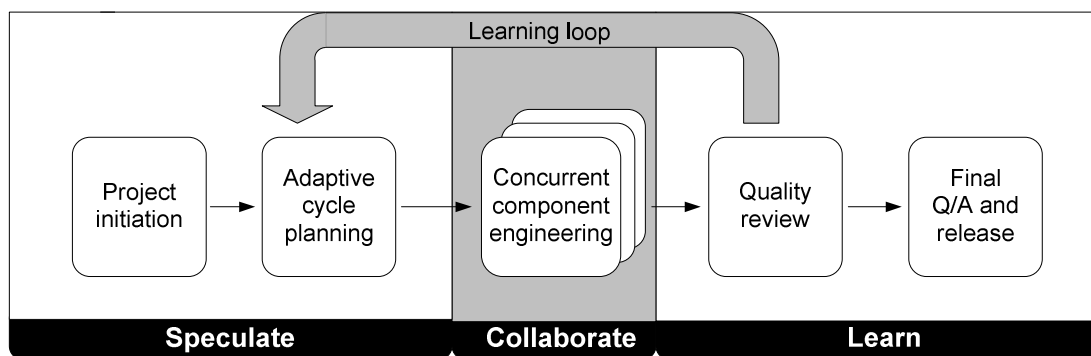
Living in an environment in which technology is ever changing, a group of developers cannot possibly know everything. Developers work together using *collaboration* skills, whereas designers make decisions or produce results (Highsmith, 2000b:24) to solve complex problems more easily. In solving large and complex problems, a large volume of information must be collected (too large for one developer), analysed and applied. The *collaborate* phase emphasises active, open, and continual communication and team collaboration as the project’s foundation (Lenz & Moeller, 2004:44; Schuh, 2004:36).

In order to become adaptive, the organisation and developers must focus on *learning*. They will then have the ability to respond to change that may occur in almost every project. “Learning

about oneself – whether personally, at a project team level or at an organisation level – can be painful” (Highsmith, 2000b:24). During this phase, the deliverables completed at each iteration are reviewed to learn from the mistakes made (Lenz & Moeller, 2004:45) in order to ensure that the same mistakes are not repeated. A post-mortem is an example of a document that could be used to determine successes and failures, but it could also be used to cast blame on a team member instead of being a learning tool. A problem is not a significant issue. Developers should learn from problems and their mistakes. Learning is an ongoing process, and mistakes are inevitable.

Highsmith (2000b:27) identifies four categories of lessons to be learnt by the end of each development cycle:

- quality of results from the client’s perspective;
- quality of results from a technical perspective;
- the functioning of the delivery team and the practices they utilise; and
- the project status.



Source: Highsmith (2000b:26)

Figure 2.13: ASD life-cycle activities

Speculate

The ASD life cycle is explained in more detail in Figure 2.13 above. In the *project initiation* phase, information is gathered to determine the scope, schedule, cost and resource requirements that are necessary to deliver the project (Lenz & Moeller, 2004:46), after which the seven adaptive cycle speculation steps are executed (Highsmith, 2000b:24):

1. Conduct the project initiation phase.
2. Determine the project time box.
3. Determine the optimal number of cycles and the time box for each.
4. Write an objective statement for each cycle.
5. Assign primary components to cycles.

6. Assign technology and support components to cycles.
7. Develop a project task list.

Most of the data should be gathered in Joint Application Development (JAD) sessions. A JAD session is a workshop in which developers and clients meet to brainstorm ideas, discuss product deliverables (features) and improve communication (Abrahamsson *et al.*, 2002:71). Initial deliverables for small projects can take a week, while large projects may require a “Cycle 0”, which involves delivering preparatory deliverables to the client, but no sections of the application (Highsmith, 2000b:26).

According to Highsmith (2000b:26), project initiation (*step 1*) involves:

- setting up the project mission and objectives;
- understanding and documenting constraints;
- establishing the project organisation and key participants;
- identifying and outlining requirements;
- making initial size and scope estimates;
- identifying key project risks.

The time box (*step 2*) should be based on the resources from project initiation, scope, requirements set by users, feature set(s), and time and budget estimates.

The individual cycle length (*step 3*) depends on two factors: the overall project schedule and the degree of uncertainty. For a small to medium-sized project, a cycle length of four to eight weeks is required.

“Each cycle should have its own theme.” (Highsmith, 2000b:26). Every cycle (*step 4*) has its own milestones to fulfil. It is important to force product visibility that will show all the mistakes, problems and defects in the project. A cycle delivers a workable presentation set of components to the client, and it makes the product visible to the development team (Highsmith 2000b:26). It is also important to remember that testing takes place continuously.

The main concern of component assignment is that every cycle should deliver a visible, acceptable result. Factors that should be taken in consideration when assigning components to cycles (*steps 5 and 6*) are (Highsmith, 2000b:26):

- ensuring that each cycle delivers something useful to clients;
- identifying and managing high-risk items early in the project;
- scheduling components to accommodate natural dependencies; and

- balancing resource utilisation.

Developers who feel uncomfortable without a task list (*step 7*) could make each component the task target (Highsmith, 2000b:27). Additional tasks with no relation to components can also be added. The primary plan of component assignment is a “component breakdown structure” and not a “work breakdown structure” (Highsmith, 2000b:27).

Collaborate

Managers are more concerned about dealing with collaboration and concurrency than the details of designing, coding and testing. In the *concurrent component engineering* activity (see Figure 2.13), the working components of the system are developed and delivered (Highsmith, 2000b:27; Lenz & Moeller, 2004:46). The working components can also be developed and delivered in parallel if more than one feature needs to be produced per cycle (Lenz & Moeller, 2004:46). Concurrency in a IT project is a critical issue. In large IT projects, concurrency could be managed by using an advanced adaptive life cycle, while in small IT projects it could be managed informally because team members work closely together. Collaborative development in small teams could be enhanced by applying some XP practices, such as collective ownership and pair programming.

Learn

In order to deliver a project of quality, review practices are executed in order to implement the lessons learnt through the *learning loop* to ensure that the same mistakes are not repeated. The following quality review practices should be applied in the learning phase (Highsmith, 2000b:27–28):

- *Providing visibility and feedback from the clients*: This can be achieved by using client focus groups, who review the application, explore a working model of the application and record client requirements.
- *Reviewing technical quality*: This focuses on the technical quality assessment of the product.
- *Monitoring the team's performance*: This can be called the people-and-process review, where post-mortems are needed. There are four basic post-mortem questions (Highsmith, 2000b:28):
 - What is working?
 - What is not working?
 - What do we need to do more of?
 - What do we need to do less of?

Post-mortems force developers to learn about themselves and explain an organisation's ability to learn.

- *Reviewing project status:* The status and progress of the project to date are assessed and reviewed. The basic questions asked for reviewing the status of a project are (Highsmith, 2000b:28):
 - What is the status of the project?
 - What is the status compared with our plans?
 - What should that status be?

At the beginning of each cycle, this review re-plans the project effort. In a component-based approach, the project status reflects multiple components at different stages of completion.

According to Abrahamsson *et al.* (2002:70), the learning loop (as seen in Figure 2.13), which is gained from repeated quality reviews, forms the basis of further cycles. The quality reviews (quality practices) demonstrate the functionality of software being developed during each cycle. Quality reviews are performed at the end of each cycle and it is important to keep clients involved using JAD sessions. In the *final Q/A and release* activity a final quality and assurance check is conducted to ensure that the gathered requirements are satisfied before the system is released into the client environment (Lenz & Moeller, 2004:46).

Characteristics of an Adaptive Software Development life cycle (Highsmith, 2000b:25)

- *Mission focused:* A mission provides boundaries rather than a fixed destination. Without a good mission statement and refinement process, an iterative life cycle will become a life cycle with no progress (oscillating life cycle). Mission statements act as guides that encourage exploration, while mission artefacts provide direction and guidance on critical decision-making (Highsmith, 2000b:25).
- *Component based:* A component of the system defines a group of features developed during an iterative cycle.
- *Iterative:* "Iterative cycles emphasize 're-doing' as much as 'doing'" (Highsmith, 2000b:25).
- *Time-boxed:* Time-boxing means setting fixed delivery times for iterative cycles and projects. Time-boxing forces a project team and clients to overlook and constantly re-evaluate the project's mission profile (consisting of scope, schedule, resources, and defects). It is about focusing on difficult trade-off decisions (Highsmith, 2000b:25).
- *Risk driven:* The adaptive life-cycle plans are driven by analysing the risks critical to the project (Highsmith, 2000b:25).
- *Change tolerant:* This is the ability to incorporate change as a competitive advantage.

Effectiveness and evaluation of Adaptive Software Development according to agile system development methodology characteristics

Adaptive Software Development does not explicitly state the business culture or technology environments in which it functions (Qumer & Henderson-Sellers, 2008b:285). It provides processes for progress tracking and functions well for small teams of four to eight people where the environment is volatile and requirements are vague (Lenz & Moeller, 2004:46–47). Unlike XP, it does not provide detailed practices (Ambler & Constantine, 2002:246), which can be viewed as a disadvantage.

Using adaptive approaches such as ASD can make large IT projects that have extreme development requirements a success by managing change and delivering an up-to-date product. Like DSDM, Scrum and FDD, ASD does not support the sixth agile characteristic value of keeping the process cost effective, meaning that it does not support leanness (an ASDM characteristic feature; Qumer & Henderson-Sellers, 2008b:285).

Adaptive Software Development presents the simplest life cycle of all the ASDMs with a life cycle. Complications may arise when complex systems are executed, as it lacks the necessary project controls to ensure that the project is well controlled in a dynamic environment. Although ASD might have the ability to learn (during the learning loop), it is more important to prevent mistakes being made without passing through the learning loop in order to ensure that a project is delivered within cost, time and quality constraints.

2.4.7 Lean Development

Lean Development (LD) was developed around 2000 by Bob Carette and is derived from the principles of lean manufacturing used during the restructuring of the Japanese automobile manufacturing industry (Toyota) in the 1980s to compete with American motor vehicle manufacturers. It was later popularised by Poppendieck (2003). Lean manufacturing still used in the automobile industry is concurrent rather than sequential. Decisions are made as late as possible with as much information as possible. Critical decisions are made by the developers themselves.

According to Ballé and Ballé (2005:18), many companies are now working at adopting 'lean' practices, because of the success Toyota's product development projects shows – with productivity that is four times better and two times faster than its US Motor Organisation equivalents. The application of lean-manufacturing principles has enabled manufacturing and service delivery organisations to "improve their competitiveness, productivity, quality and client service" (Riezebos, Klingenberg & Hicks, 2009:237).

Just like other ASDMs, LD focuses on clients (people), iterative development, value flow and accelerating application development, but not at the expense of higher defect or cost rates (Highsmith, 2002a:6). It provides a toolkit of guiding principles that determine the maturity of an organisation by how reliably and rapidly it can serve its clients (Schuh, 2004:40). According to Cohen *et al.* (2003:18), other ASDMs try to change the development process. To be truly agile, LD should change the way organisations work from the top down. Lean Development is a management philosophy rather than a development process. According to Highsmith (2002c:290), “LD embodies the concept of dynamic stability”, which means that it has the ability to adapt rapidly to client expectations while building stable and continuous improved processes that are flexible across different products.

Using LD as an ASDM, the key for any organisation is to be more agile than their competitors. This means an organisation that is change tolerant and focuses on risk entrepreneurship. The latter enables companies to turn risk into opportunity and to use the opportunity to their own benefit. A change-tolerant organisation is one that remains competitive in a changing environment (Highsmith, 2002a:6). Every business should develop into a highly tolerant organisation in order to cope with changes. According to Highsmith (2002a:6), there are three main goals of LD.

Complete the project in:

- one-third of the time;
- one-third of the budget; and
- one-third of the defect rate

than those of previous projects.

To be lean, the organisation has to think lean. According to Poppendieck (2003:2), “lean thinking” entails allowing clients to “delay their decisions about exactly what they want as long as possible, and when they ask for something, give it to them so fast they don’t have time to make up their minds”.

There are seven guiding principles of LD. Poppendieck (2003) provides 22 tools for converting lean principles into agile system development practices. The seven lean principles are discussed below and their associated tools are mentioned.

The seven Lean Development principles

1. Eliminate waste

Anything that is not of value to the client or system is seen as waste (Poppendieck &

Poppendieck, 2003:xxv; Schuh, 2004:41; Dahlke, 2008:13). If waste is identified, it should be eliminated immediately. There is no reason for developing endless packs of documentation that will never be read (Phillips, 2002:178). The seven wastes in software development are (Poppendieck, 2003:3; Poppendieck & Poppendieck, 2003:4):

- partially completed work (the “inventory” of a development process);
- additional features (develop only what clients want at the present moment);
- additional processes (easy to find when documentation is viewed as important);
- handoffs (much tacit knowledge gets lost);
- task swapping (team members should do one thing at a time);
- defects (at least defects that are not quickly caught by a test); and
- waiting (for information and orders).

Lean Development focuses on the elimination of waste by looking at the flow of value from the request to implementation. In order to facilitate value flow, teams should be formed to implement each requirement from start to finish as quickly as possible.

Tools: Seeing waste, value stream mapping (Steindl, 2004; Norton, 2005).

2. Amplify learning

“Great designers understand that designs emerge as they develop a growing understanding of the problem.” (Poppendieck, 2003:2). Amplified learning differentiates between the production process (reducing variation) and development process (defining the right system; Schuh, 2004:41). Development is a learning process, and learning is crucial to the success of a project (Dahlke, 2008:14). Developers discuss what should be done, ways to make it work and they try it. If it does not work, they learn from their mistakes and try again.

The idea is to adapt to variation by constant feedback to clients and not to eliminate variety as a whole. Because of iterative development, developers can measure the difference between what the client wants and what the software can do in order to make the correct adjustments. Lean Development focuses on feedback, using short (a week to a month) and full cycle (tested, integrated and deployed code) iterations. According to Poppendieck (2003:4), “Iterative (evolutionary) development is the best approach for software development.”

Tools: Feedback, iterations, synchronisation, set-based development (Steindl, 2004; Norton, 2005).

3. Delay commitment

Delaying commitment means allowing for all options for as long as system development allows it. The reasons for having delayed commitments in complex projects are to build capacity for possible change (Poppendieck & Poppendieck, 2003:xxvi) and “to wait until the future is closer and easier to predict” (Schuh, 2004:41). The fundamental concept of LD is delaying “irreversible decisions until they can be made based on known events rather than forecasts” (Poppendieck, 2003:4), because the later a decision is made, the more informed it will be (Dahlke, 2008:15). Having options forces clients to delay decisions until they have sufficient and accurate information to make the correct (not predicted) decision.

The following are ways to allow for all options in system development (Poppendieck, 2003:5):

- share development information that is partially complete;
- organise for direct, person-to-person collaboration;
- determine when decisions must be made;
- determine the manner in which changes must be accepted;
 - separate concerns;
 - circumvent repetition;
 - encapsulate variation;
 - postpone implementation of future capabilities;
- agree on and commit to refactoring; and
- make use of automated test software.

Tools: Option thinking, the last responsible moment, making decisions (Steindl, 2004; Norton, 2005).

4. Deliver quickly and fast

The goal here is to create value as quickly as possible, once the client has decided what he/she wants. This means (Poppendieck, 2003:5):

- no delay in deciding which requests to approve;
- no delay in staffing and immediate clarification of requirements;
- no time-consuming handoffs;
- no delay in testing;
- no delay in integration; and
- no delay in deployment.

In mature software development organisations, “all of this happens in one smooth, rapid flow” in response to client requirements (Poppendieck, 2003:5). If the project team does not deliver quickly, it will not have the ability to delay decisions or to provide reliable feedback to the client (Poppendieck & Poppendieck, 2003:xxvi). The focus should be on delivering what is required at the present moment, instead of predicting what may be required later (Schuh, 2004:41).

Tools: Pull system, queuing theory, cost of delay (Steindl, 2004; Norton, 2005).

5. Empower the team

In LD, the team makes its own process designs, commitments, goals and decisions on the manner in which to complete these specified goals. A team can only be allowed to make its own decisions if it is empowered through expertise, training and leadership. The developers must be involved in technical decisions to ensure that what was promised to the client can be delivered by the system (Poppendieck & Poppendieck, 2003:xxvi).

Once a team is empowered, it can make better decisions. It is management's responsibility to supply the project teams with the necessary training, expertise and leadership (Phillips, 2002:178), as well as information to make the best decisions in order to deliver a successful project. By working directly with clients in order to understand their requirements and working with other developers in order to determine that these requirements can be satisfied, results can be presented frequently to clients to determine whether development processes are on track.

Tools: Self-determination, motivation, leadership, expertise (Steindl, 2004; Norton, 2005).

6. Build in integrity

Two kinds of integrity exist, namely perceived integrity and conceptual integrity. Perceived integrity is exactly what the client and users wanted (Dahlke, 2008:16), although they did not ask for it. In this case the client will be the organisation providing the system development expectations, while the users are the people that will be using the system to assist them in completing their daily responsibilities. In order to achieve perceived integrity, the organisation should have continuous and detailed communication and a flow of information from the users to the developers (Phillips, 2002:178). This is achieved where the master designer (architect) understands the detail of the domain and ensures that developers always have user requirements at hand to make correct decisions that will be of value to clients.

Conceptual integrity presents software to clients with a single consolidated explanation of the manner in which the tasks are to be completed in order to satisfy their requirements. “Conceptual integrity means that all parts of a software system work together to achieve a smooth, well functioning whole.” (Poppendieck, 2003:6). The flow of detailed information between team members and technical members of a project will achieve conceptual integrity within a balanced system. Everyone, from supplier to client, should be involved in the progress of development from the beginning of the project. Many believe that integrity comes from a documentation-centric approach, but according to Poppendieck (2003:7), organisations should rather use a test-centric approach. Furthermore, she states that organisations should “test early, test often, test exhaustfully, and make sure an automated test suite is delivered as part of the product”. This principle implies that the system is well designed, user friendly, runs without errors, and is easy to maintain and to extend (Schuh, 2004:41).

Tools: Perceived integrity, conceptual integrity, refactoring, testing (Steindl, 2004; Norton, 2005).

7. See the whole

Overall success of the project is more important to LD than the traditional sub-optimisation of individual tasks. The success of a whole system is based on its parts being well integrated to work together – not the quality of the individual system parts (Schuh, 2004:41). When change is required to any individual part of the system the effect of the change on the whole system is considered (Dahlke, 2008:16). The project team must take caution not to only focus on the quality of individual system parts (Phillips, 2002:179).

The most appropriate way to encourage collaboration and avoid sub-optimisation is to make the team accountable for their control and influence. This means measuring the team’s performance and defect count, and not that of the individual team member. It may seem unfair to hold the whole team accountable for an individual team member's performance but this causes teams to work together, address problems amongst themselves, and assume responsibility to plan their own processes.

Tools: Measurements, contracts (Steindl, 2004; Norton, 2005).

According to Highsmith (2002a:6), Carette's LD sends three messages to developers using ASDMs:

- "The wide adoption of ASDEs (Agile Development Software Ecosystems) will require strategic selling at senior levels within organisations."
- "The strategic message that will sell ASDE's is the ability to pluck opportunity from fast-moving, high-risk exploration situations."
- "Proponents of ASDEs must understand and communicate to their clients the risks associated with agile approaches and, therefore, the situations in which they are and are not appropriate."

Maturity (deliver quickly) is measured by operational excellence, that is, the speed with which clients can be served repeatedly and reliably (Poppendieck, 2003:3). Maturity is not measured by the manner in which plans are followed or the "comprehensiveness of process documentation" (Poppendieck, 2003:3).

Effectiveness and evaluation of Lean Development according to agile system development methodology characteristics

The lean principles identified by Poppendieck (2003:3–7) have lead to significant improvements in several areas such as health-care delivery, logistics, building construction, product development and military logistics. LD has also been successful in a number of large telecommunication projects in Europe (Highsmith, 2002a:6). Its degree of agility cannot be examined here, as it was not investigated by Qumer and Henderson-Sellers (2008b:280) and is not otherwise reported on in the literature. It can however be predicted that LD, like DSDM, Scrum, FDD and ASD, does not support the sixth agile characteristic value of keeping the process cost effective and thus does not support leanness (an ASDM characteristic feature. Lean Development can make traditional SDMs more agile if they are combined (Schuh, 2004:37).

What can be said is that LD draws from the success of lean manufacturing (Stamelos & Sfetsos, 2007:14) and is based on the underlying understanding of lean thinking (Poppendieck, 2002; Larman & Vodde, 2009), which proved to be very effective in production, sales, service and human resource environments. Because LD is still relatively new to the world of software development, the effectiveness of LD must still be evaluated in IT environments in order to determine how effective LD really is.

2.5 The effectiveness and acceptance of agile system development methodologies in practice

Agile system development methodologies are gaining popularity and many organisations have adopted ASDMs since the creation of ASDMs in the early 1990s (Good, 2003:27). Since 2001, the average number of conference papers and general articles per year on agile software development has increased (Dingsoyr, Dyba & Moe, 2010:4), which shows that there is growing awareness of ASDMs. Agile system development methodologies can also be combined with each other, of which the XP/FDD and XP/Scrum combinations are the most popular (McMaster, Wastell, Fernely & DeGross, 2007:242).

Organisations are increasingly adopting ASDMs within their projects (Good, 2003:28). The effectiveness of ASDMs is still being studied, but the functionality in certain circumstances seems promising, and deserves the attention of systems developers (Mendonca, 2002:505; Ambler, 2002:9).

Application area

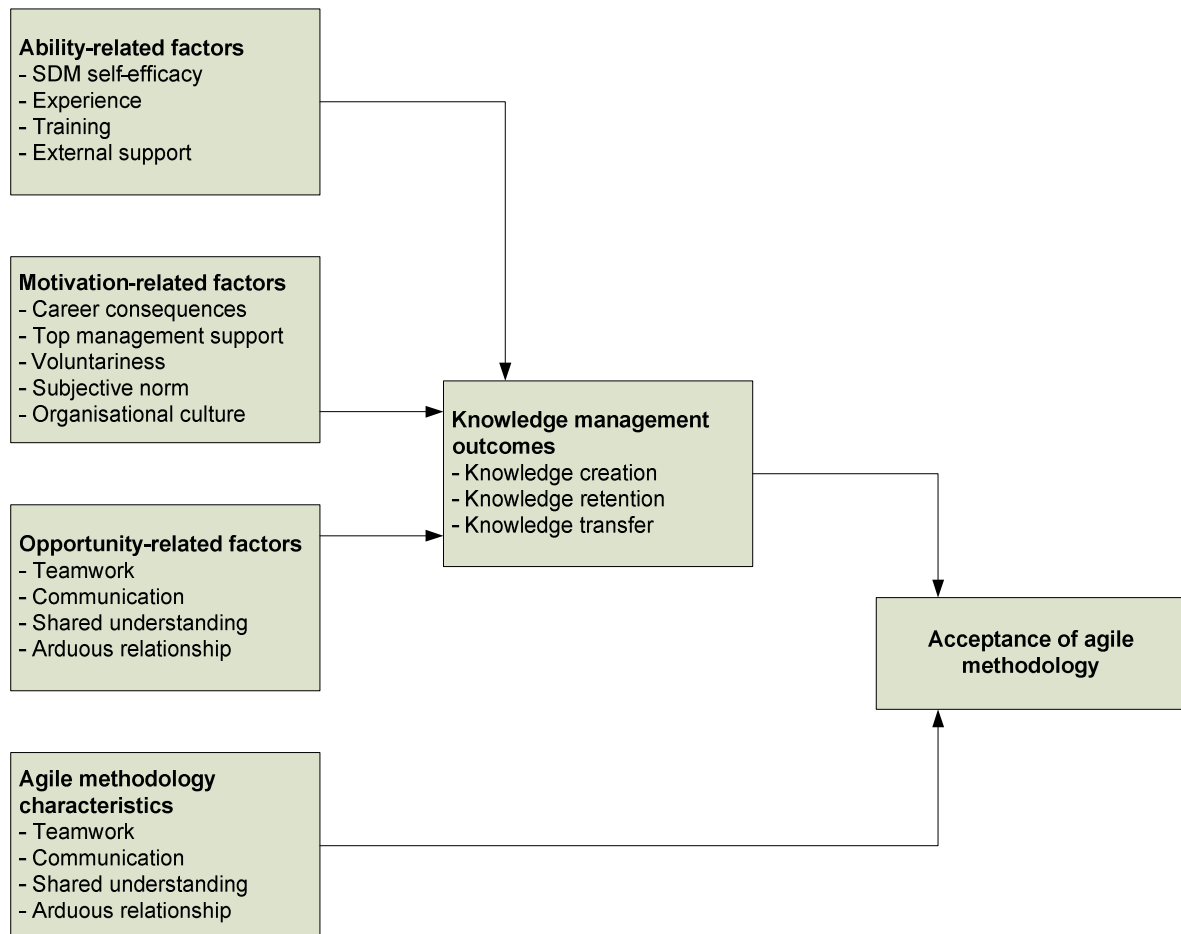
When implementing an ASDM into an organisation, there are several factors under management control that significantly affect the success of the implementation (Livermore (2008:35). These factors include:

- training on the ASDM;
- active management involvement and support;
- access to external resources; and
- organisation size (large, medium, small).

Currently, there are more documented success stories about small projects than large projects (Lindvall *et al.*, 2002:206; Cohen *et al.*, 2003:31). According to Liu and Roussev (2006:111), ASDMs “can be applied only to small-sized projects”. This could be because managing a large IT project is much more difficult. Peterson and Wohlin (2009:1488) explain that new issues relating to increased complexity when scaling ASDM projects arise when using ASDM in projects. Myer (2008:57) states that “most ASDMs are adaptive, but they only work within certain contexts”. He explains that team size must be limited, the team must be able to organise itself and communicate daily, and the organisation’s governance framework will have to adapt and change.

In order to overcome the challenge of ASDM acceptance in organisations, Chan and Thong (2009:803) developed a conceptual framework for ASDM acceptance. This conceptual

framework had the ability to assist practitioners with the effective and successful deployment of ASDMs in organisations. It may also provide guidance to researchers regarding the acceptance of ASDMs in practice. Their framework is developed from a knowledge-management perspective and can be seen in Figure 2.14. The ability-related, motivation-related, opportunity-related, and knowledge-related factors are relevant to knowledge management within systems development to understand the acceptance of ASDMs. The characteristics of ASDMs complement the conceptual framework by “adding the long-standing technology adoption perspective” (Chan & Thong, 2009:807).



Source: Chan and Thong (2009:807)

Figure 2.14: A conceptual framework for ASDM acceptance

While Chan and Thong’s conceptual framework assists with ASDM acceptance in organisations, Qumer and Henderson-Sellers (2008a:1899) found that there are few organisations in practice that can adopt ASDM approaches fully in a short period, as full transition normally takes several years. In order to overcome this challenge, the ASSF was developed by Qumer and Henderson-Sellers (2008a:1899–1919), which offers an approach to the investigation and examination of agile knowledge and governance. The ASSF contains an agile toolkit, which quantifies and tailors software processes by constructing and evaluating

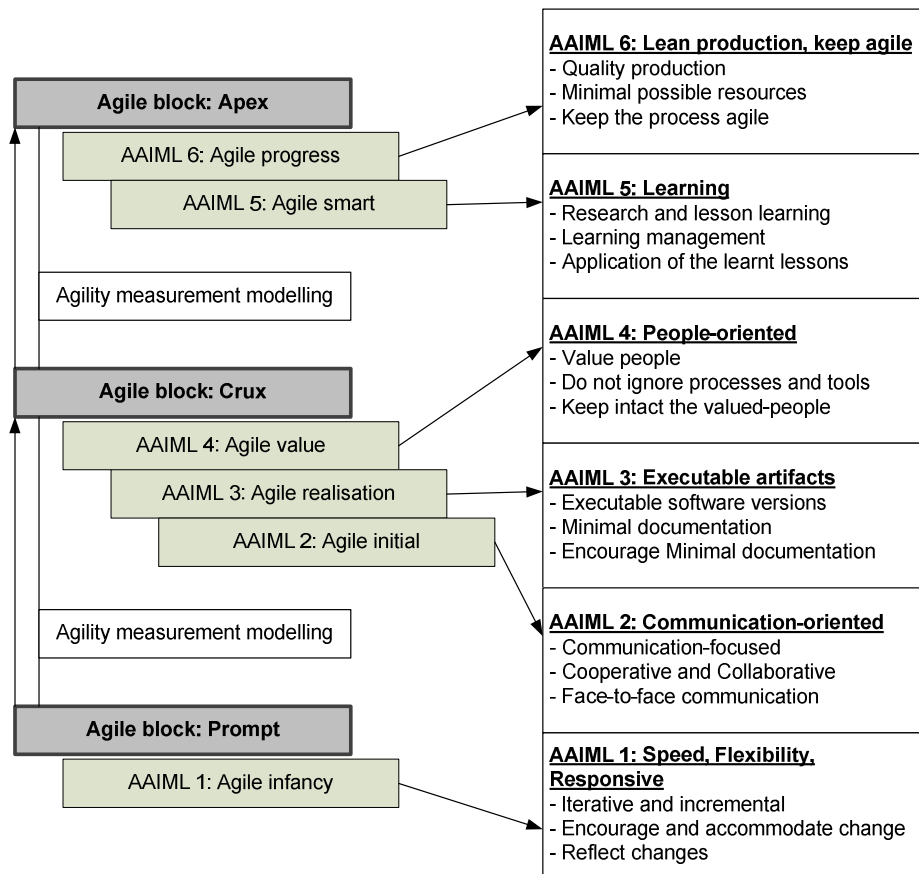
multi-abstraction (such as object-orientated, etc.) “situation specific process fragments or processes for the development of complex software development projects that may involve more than one abstraction mechanism” (Qumer & Henderson-Sellers, 2008a:1901). The ASSF also contains 4-DAT, also known as the agility calculator – a component of the agile toolkit, which was first mentioned in Section 2.3. The 4-Dimensional Analytical Tool is used to “analyse and quantitatively measure the agility of any software development methods or approaches including governance, knowledge engineering and management” (Qumer & Henderson-Sellers, 2008a:1904). The 4-Dimensional Analytical Tool measures agility using four dimensions as presented in Table 2.1.

Table 2.1: Four dimensions of 4-DAT

4-DAT's four dimensions	
Dimension 1 (method scope)	
SCOPE	DESCRIPTION
1. Project size	Does the method specify support for small, medium or large projects? (business or other?)
2. Team size	Does the method support for small or large teams? (single or multiple teams?)
3. Development style	Which development style (iterative or rapid) does the method cover?
4. Code style	Does the method specify code style (simple or complex)?
5. Technology environment	Which technology environment (tools, compilers) does the method specify?
6. Physical environment	Which physical environment (co-located or distributed) does the method specify?
7. Business culture	What type of business culture (collaborative, cooperative or non-collaborative) does the method specify?
8. Abstraction mechanism	Does the method specify an abstraction mechanism? (object-oriented, agent-oriented?)
Dimension 2 (Agility Characterisation)	
FEATURES	DESCRIPTION
1. Flexibility	Does the method accommodate expected or unexpected changes?
2. Speed	Does the method produce results quickly?
3. Leanness	Does the method follow the shortest time span, use economical, simple and quality instruments for production?
4. Learning	Does the method apply updated prior knowledge and experience to create a learning environment?
5. Responsiveness	Does the method exhibit sensitiveness?
Dimension 3 (Agile Value Characterisation)	
AGILE VALUES	DESCRIPTION
1. Individuals and interactions over processes and tools	Which practices value people and interaction over processes and tools?
2. Working software over comprehensive documentation	Which practices value working software over comprehensive documentation?
3. Customer collaboration over contract negotiation	Which practices value customer collaboration over contract negotiation?
4. Responding to change over following a plan	Which practices value responding to change over following a plan?
5. Keeping the process agile	Which practices help in keeping the process agile?
6. Keeping the process cost effective	Which practices help in keeping the process cost effective?
Dimension 4 (Software Process Characterisation)	
PROCESS	DESCRIPTION
1. Development process	Which processes cover the main life cycle process and testing (quality assurance)?
2. Project management process	Which practices cover the overall management of the project?
3. Software configuration control process/ support process	Which practices cover the process that enable configuration management?
4. Process management process	Which practices cover the process that is required to manage the process itself?

Source: Qumer and Henderson-Sellers (2008a:1904)

Like any organisational change, ASDMs creates uncertainty and confusion when an organisation incorporates it (Beyer, 2009:9) which in turn creates a resistance to change that must be managed. In order to assist organisations in the transition to an ASDM, Qumer and Henderson-Sellers (2008a:1916–1917) introduced an Agile Adoption and Implementation Model (AAIM) that thus far was found to be successful for large non-agile software development organisations. The Agile Adoption and Implementation Model consists of three agile blocks (Prompt, Crux, Apex) and six agile stages as seen in Figure 2.15. The 4-Dimensional Analytical Tool is used to measure the degree of agility within each block quantitatively. Each stage has a specific set of agile practices that must be followed to achieve a specific AAIM level.



Source: Qumer and Henderson-Sellers (2008a:1908)

Figure 2.15: The Agile Adoption and Implementation Model

Surveys completed by practitioners indicated positive results when organisations utilised ASDMs. In 2002, Reifer (2002:16–17) surveyed thirty-two development organisations, of which fourteen were using ASDMs on thirty-one individual projects. The result of his study was that seven of the fourteen organisations that used ASDMs captured hard cost, productivity and quality data. Five of the seven had benchmarks that they could use for comparison purposes (Reifer, 2002:17). He came to the following conclusions on organisations using ASDMs (Reifer, 2002:17):

- *Productivity improvement:* 15 to 25% average gain based on published industry benchmarks.
- *Cost reduction:* 5 to 7% on average based on published industry benchmarks.
- *Time-to-market compression:* 25 to 50% less time compared with previous projects in participating companies.
- *Quality improvement:* Five companies had data showing that their defects rates were on par with their other projects when products or applications were released.

A global survey carried out by an Australian organisation presented the following findings about the utilisation of ASDMs by organisations (Good, 2003:27–28):

- 88% of organisations cited improved productivity;
- 84% reported improved quality of software production;
- 46% of respondents reported that development costs were unchanged using ASDMs, while 49% stated that costs were reduced or significantly reduced;
- 83% stated that business satisfaction was higher or significantly higher;
- 48% cited that the most positive feature of agile methodologies was their ability to respond to change rather than follow a predefined plan.

An electronic workshop was held by Scott Ambler (independent consultant specialising in object-oriented development), Barry Boehm (well-known for his spiral software development life cycle and COCOMO II estimating technique), Kent Beck (originator of XP), Alistair Cockburn (author of *Agile Software Development*) and Randy Miller (co-author of *Advanced Use Case Modelling*) was held to discuss the following three points regarding ASDMs (Ambler, 2002:9):

1. Agile development works better for smaller teams (up to twenty or thirty members).
2. Agile system development methodologies work best when the future is unknown (designed for current, not future needs).
3. Agile system development methodologies suit applications that can be built quickly and do not require extensive quality assurance, and they work less well for critical, reliable and safe systems.

Before implementing an ASDM or starting a transition process to achieve acceptance of a certain ASDM in an organisation, Lindvall *et al.* (2002:206) provided some lessons learnt, which must be taken into consideration:

- Experience in agile projects is very important for it to succeed, although experience in the actual building of the system is more important. It is estimated that 25 to 33% of project staff must be experienced, but in cases in which pair programming is practised in teams in which they monitor each other, it might be as low as 10%.
- Agile system development methodologies can be used to conduct safe, critical and reliable projects. Critical issues are easily addressed in ASDMs because clients give requirements, state the importance of each requirement and provide input through system development. The key is that the performance requirements must be made explicit early on in the project and that proper levels of testing be planned.
- Agile system development methodologies need less formal training than traditional SDMs. Training is minimised by the fact that pair programming is used. This is more important than regular training because of the experience gained from learning from one another. There is

training material available for XP, Scrum, FDD and Crystal ASDMs.

- The most important success factors are culture, people and communication. Agile system development methodologies need cultural support, otherwise the methodology applied will not succeed. They use fewer but more competent people than traditional SDMs. Communication is enhanced by using pair programming and through constant interaction with clients who give frequent feedback.
- By using ASDMs in projects, warning signs can be detected early in project development. Warning signs include low interest at meetings and production of “useless documentation”.
- Refactoring should be done on a regular basis with code of reasonable size, keeping the scope down and local. Agile system development methodologies make large-scale refactoring more feasible than traditional SDMs. If a set of automated tests is maintained, changes to large architectural designs do not have to be risky.
- Documentation makes the design “heavy” and should be assigned as a cost. Organisations normally ask for more than is needed. In order to give value and satisfy requirements, the main goal should be communication, and useless documentation should be avoided.

Many organisations have embraced ASDMs to encourage innovation and reduce development time and risk (Kloppenborg, 2009:233). Even the United States Department of Defence specifies that ASDMs will be used in their software development (Good, 2003:29). Highsmith (2002a:9) agrees with Ambler (2002:9) that “agile processes are ‘real’: they’re here to stay and every IT professional needs to take them seriously”. According to Beyer (2010:53), ASDMs do extremely well by taking a product defined in detail with restricted scope to produce valuable software quickly, but if a large-scale project is done, planning, research and design work must be done beforehand in order to deliver a system of value. Pence (2007:38) argues that ASDMs minimise some fundamental concerns, in that they do not encourage upfront design or client contracts. He advises that ASDMs be introduced to an organisation with “political tact and care” because its success depends on the entire team believing in it.

Advantages

In general, there is little information about the effectiveness of some of the well-established ASDMs discussed in this literature review. Some of them were only tested by the authors themselves, but a methodology truly works when other developers test it and find it to be of value. In recent years, there has been more intensive analysis of the effectiveness of ASDMs, but gathering hard empirical data on ASDMs is difficult because of the nature of software production (Good, 2003:29). The advantages of using ASDMs include (Goodpasture, 2010:24):

- Business milestones are more strongly committed to.
- Adapting efficiently to changing stakeholder requirements and priorities will ensure that the

project stays current and relevant.

- Regular and rapid deliveries on production offer early benefits and the possibility of the project supporting itself financially.
- For teams that have up to 25 developers, it can have the financial benefit of being cost effective.
- As the project progresses, clients have the opportunity to influence the value proposition of the project.
- As small teams work collectively on client requirements, the creative potential of the project team is unleashed.
- The successful completion of an iteration is celebrated in order to generate a sense of accomplishment amongst the team members.
- The objective of the project is client-centric and not necessarily bound to an outdated plan.
- The value provided to clients is validated almost automatically because of the manner in which ASDMs are designed.
- Trust is won by delivering on the stakeholder and client expectations and requirements.

The strengths of the ASDMs were used in the development of the hybrid APMM (ver. 0), which will be reported on in Chapter 5.

Disadvantages

Organisations do not actually report on whether the ASDMs they used have failed. In addition, in the case of failures, it should be determined whether the methodology chosen was used correctly according to the specifications (for example principles, practices, rules and guidelines). If not, organisations and role-players cannot say the chosen ASDMs failed.

Ironically, as failures are largely not reported developers are given assurance that ASDMs can deliver IT projects successfully. Agile system development methodologies have some defects, which can only be identified by people using these methodologies (excluding the authors), who will test them in different environments. Shelly and Rosenblatt (2010:145) explain that projects that follow ASDMs can be exposed to constant scope change because of changing requirements. They furthermore emphasise that teams require advanced interpersonal and technical skills and that a lack of structure and documentation might present risk factors that cannot be ignored.

There is no perfect methodology that will work in all kinds of environments and under all circumstances. The disadvantages of ASDMs include (Goodpasture, 2010:25):

- There is the possibility of a high turn-over of staff on the team.
- Many dependencies may be discovered later on in the project because ASDMs are not

architecture driven.

- It is difficult to scale the dynamics of a small team to an enterprise-scope project.
- It is difficult to scale a project without having formal commitments to documentation.
- Commitment to scope and cost is poor.
- Because the scope requirements are not adequately known, it is not an easy task to contract or procure the work team.
- Running a project using ASDMs depends on face-to-face communication, a group of skilled and talented staff members, immediate access to well-informed and empowered clients and favourable logistics for team collocation.
- Testing is not conducted separately.
- Product and deliverable verification is done by testing and not traceable to specifications.
- A highly reliable and mission-critical project requires definitive verification that is lacking when applying ASDMs.

As with the advantages of ASDMs, their weaknesses were addressed in the development of the hybrid APMM (ver. 0), which will be reported on in Chapter 5.

These results demonstrate a very positive attitude towards ASDMs and many significant software development companies wish to implement them. Highsmith (2002a:9) states that large companies in countries such as the USA, Australia, Europe and India found ASDMs to be very effective because a product could be produced in a “turbulent, ever-changing, ever-existing marketplace”. Many companies have started to utilise ASDMs, and a number of large software clients have begun to demand that their software be developed using ASDMs. Organisations cannot ignore ASDMs because they have much to offer, but organisations that use traditional SDMs can face several challenges when adapting ASDMs (Nerur, Mahapatra & Mangalara, 2005:73). Before implementing an ASDM, organisations should consider Qumer and Henderson-Sellers’ (2008b:280) analytical framework, 4-DAT, for assistance when evaluating the agility of an ASDM before it is implemented.

2.6 Comparison of the seven agile system development methodologies

The seven ASDMs described in this study will be compared using the four elements of an SDM (Huisman & livari, 2006:32), as explained in Section 2.2 to emphasise some of the main similarities and differences amongst the different ASDMs. The comparison of the different ASDMs is summarised in Table 2.2.

Table 2.2: Comparison of ASDMs

		SDM elements			
		System development approach	System development process model	System development method	System development technique
ASDMs	DSDM	Customer-centric and agile approach	Incremental and iterative model	• Pre-project • Feasibility and business study • Functional model integration • Design and build iteration • Implementation • Post-project	• Agile – select preferred techniques
	Scrum	Customer-centric and agile approach	Incremental and iterative model	• Product backlog • Sprint backlog • Sprint • 15-minute meetings • Present functionality	• Scrum meetings; Agile – select preferred techniques
	XP	Customer-centric and agile approach	Incremental and iterative model	4 values yield 15 principles and 4 basic activities form the basis of the 12 core practices, which are incorporated into the XP lifecycle: • <i>User stories</i> • <i>Architectural spike</i> • <i>Release planning</i> • <i>Spike</i> • <i>Iteration</i> • <i>Acceptance tests</i> • <i>Small releases</i>	• Story cards • 12 core practices (pair programming, etc.) • Agile – select preferred techniques
	FDD	Customer-centric and agile approach	Incremental and iterative model	• Develop an overall model • Build a feature list • Plan by feature • Design by feature and build by feature	• Subject area breakdown into features • Agile – select preferred techniques
	Crystal ASDMs	Customer-centric and agile approach	Incremental and iterative model	• CC • Crystal Yellow • CO • Crystal Red, etc.	• Agile – select preferred techniques
	ASD	Customer-centric and agile approach	Incremental and iterative model	• Speculate • Collaborate • Learn	• JAD sessions Learning loop • Agile – selected preferred technique
	LD	Customer-centric and agile approach	Incremental and iterative model	• 7 LD principles and 22 tools	• Delayed commitment • Agile – selected preferred techniques

In Table 2.2 above, there is a clear similarity amongst all ASDMs in that they follow an agile and customer-centric system development approach and incremental and iterative system development process model. The different ASDMs have their own system development

methods, while the different techniques and tools can be chosen depending on whichever is preferred and relevant to the surrounding project environment. Some unique techniques incorporated into each ASDM are presented in the system development technique column in the table. Other similarities and differences can also be viewed in the core ideas and scope of use columns in Table 2.3, in which a summary of the seven ASDMs is presented.

The similarities and differences of PMMs will be presented in detail in Chapter 3, as this is relevant to the identification of the key strengths that were used to develop the hybrid APMM (ver. 0) in Chapter 5. The key strengths of each ASDM have already been explained in the effectiveness and evaluation section of each ASDM and the key strengths for ASDMs in general have already been explained in Section 2.5, which will be used to develop the hybrid APMM (ver. 0).

Each ASDM has its own strengths and weaknesses, which emphasises that an individual ASDM is not as efficient as using a combination of the different strengths of each ASDM. There is thus a need to combine the different ASDMs with each other and with other methodologies, such as PMMs, to increase the ability to deliver IT projects in a constantly changing business and project environment.

2.7 Summary

In this chapter, an SDM was first defined using four main aspects identified by Huisman and livari (2006:32). After defining an SDM, an ASDM could be defined and the seven core ASDMs that are relatively well established in today's environment were discussed (Highsmith, 2002a:6). During the explanation of the seven ASDMs, it became clear that ASDMs focus on people, incremental development and communication. A summary of the seven core ASDMs is given in Table 2.3.

Table 2.3: Summary of ASDMs

	Time and Author	Core ideas	Scope of use	Current status
DSDM	Defined by the 16 founding members of the DSDM Consortium in 1994.	Applications of controls to RAD. Emphasises time and resources.	Team size between 2 and 6. Multiple teams exist. Can be used in large project, if the system can be split into components.	Widely used in UK. E-DSDM was released in 2001.
Scrum	Developed by Ken Schwaber and Jeff Sutherland in 1995.	Does not prescribe specific practices, but needs management practices and tools.	Small teams with <10 members.	Ongoing research aims to integrate Scrum and XP.
XP	Created by Kent Beck in 1996 and later published in his book in 1999.	12 key practices (such as refactoring, test before coding). No process to fit every project.	Small and medium-sized teams with 3-20 members.	Growing. More practical experience than academic research.
FDD	Created by Jeff De Luca and Peter Coad in 1997.	Focuses on design and building phase. Emphasises iterative development. Needs other supporting approaches.	Claims to be suitable for the development of large software project.	Some consulting firms advocate it.. Relatively new and still evolving.
Crystal	Developed by Alistair Cockburn in 1999.	A set of methodologies. Suggests development cycle within 4 months. Emphasis on communication. Allows adoption of other ASDMs.	Not good for life-critical system. Up to 40 persons in local development.	4 proposed Crystal ASDMs. 2 exist.
ASD	Created by Jim Highsmith and first documented in his book in 2000.	Emphasis on incremental, iterative development.	Focuses on developing large projects. No built-in limitation.	No significant research.
LD	Created by Bob Carette during the 1980s, later popularised by Poppendieck's book in 2003.	7 principles used as guideposts with 22 tools.	Highsmith (2002a:6) states LD is effective in large projects	Used in large telecommunication, health, logistics, military and construction projects.

Source: Abrahamsson *et al.* (2002:18–73)

Each ASDM's effectiveness was described and the individual ASDMs was evaluated according to the agile characteristic features and values as provided by Qumer and Henderson-Sellers (2008b:280). A summary of the evaluation is given in Table 2.4.

Table 2.4: Evaluation of ASDMs towards ASDM characteristic features and values

		ASDMs						
		DSDM	Scrum	XP	FDD	Crystal	ASD	LD
ASDM features	1. Flexibility	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	2. Speed	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	3. Leanness	No	No	No	No	Yes	No	No
	4. Learning	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	5. Responsiveness	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ASDM values	1. Individuals and interactions valued above processes and tools	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	2. Working software valued above comprehensive documentation	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	3. Client collaboration valued above contract negotiation	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	4. Responding to change valued above following a plan	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	5. Keeping the process agile	Yes	Yes	No	Yes	Yes	Yes	Yes
	6. Keeping the process cost effective	No	No	No	No	Yes	No	No

Source: Qumer and Henderson-Sellers (2008b:282-288)

After discussing the individual ASDMs and their effectiveness, ASDMs in general were evaluated by considering their area of application, advantages and disadvantages. A comparison was then done of the seven different ASDMs using the four elements of an SDM (Huisman & livari, 2006:32) to emphasise some of the main differences and similarities amongst the different ASDMs.

Agile system development methodologies give developers the ability to adapt to an ever-changing environment to produce a product that is of value and current. Agile system development methodologies have been used with varying success by organisations in their IT projects. There is no unique ASDM that will work in all circumstances. Some ASDMs work only in small IT projects, while others work only in large IT projects. Furthermore, there are ASDMs that might work in both large and small IT projects. The effectiveness of ASDMs is still under study, but that they are in common use has been established.

It was reported in the chapter that the evidence supports the conclusion that ASDMs can deliver software on time, within budget, and in a constantly changing environment if implemented correctly. Although ASDMs can deliver, their disadvantages emphasise that ASDMs alone are not sufficient to solve project-management problems, which could cause a project to fail. For this reason, this study sought to determine whether ASDMs and PMMs could be combined to deliver IT projects in a constantly changing project and business environment. It also sought to further evaluate the ability of ASDMs to deliver IT projects effectively in a constantly changing environment. For this reason, PMMs will be examined in the next chapter.

CHAPTER 3

PROJECT MANAGEMENT METHODOLOGIES

3.1 Introduction

In this chapter, project management methodologies (PMMs) will be investigated. The term “project management” will first be defined, after which three PMMs will be discussed in detail because regular reference will be made to specific areas of the different PMMs when the development of the hybrid APMM (ver.0) is explained in Chapter 5. The first PMM that will be studied is the fourth edition of PMBOK (Project Management Body of Knowledge), an internationally recognised standard and project management guide, which was published by the Project Management Institute (PMI), which explains the fundamentals of project management for almost any type of project. The second PMM that will be explained is the fifth edition of PRINCE2 (PProject IN Controlled Environment), which is a registered trade mark of the Office of Government Commerce (OGC). This PMM was originally developed as a UK Government standard for information technology (IT) projects in particular. The third PMM is the relatively new APM (Agile Project Management) that has the ability to adapt to a changing project environment. Because the new editions of these PMMs were only released recently there is a limited amount of literature pertaining to the updated editions and thus reference will be made only to established sources in the explanation of each PMM.

The reason for selecting these three PMMs is that PMBOK and PRINCE2 are internationally recognised PMMs with a track record for delivering all types of projects, including IT projects for more than twenty years. Agile Project Management is appropriate, as it represents the main focus for this study, to develop a hybrid APMM, and it is part of a fresh approach towards managing projects. Each of these PMMs will be evaluated by looking at their area of application, advantages, disadvantages, and integration with other methodologies.

In the final section of this chapter, the similarities and differences of these PMMs will be discussed. This comparison served to determine gaps in these PMMs, which was used to strengthen the hybrid APMM that was built in this study and will be reported on in Chapter 4.

3.2 Definition of “project management”

“Project management” is a widely used term in practice today and before defining the term it is important to understand its origin, what it entails, and what a project is. Figure 3.1 presents the history of project management from 1950 to the latest editions of modern PMMs used in practice today.

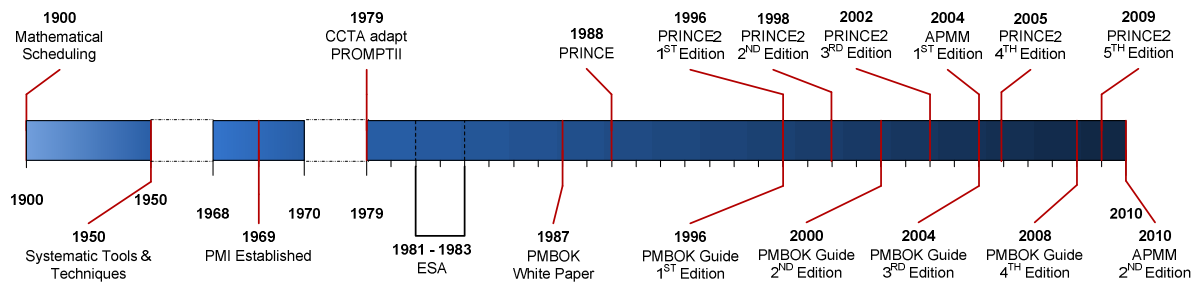


Figure 3.1: A historical timeline of project management

The term “project management” was developed in different fields (construction, engineering, architecture, etc.). Some of the first investigators of project management were Henry Laurence Gantt (1861–1919) and Henri Fayol (1841–1925). Gantt developed the well-known Gantt Chart that is still used in practice today, while Fayol developed the primary functions and principles of management, which are still applied in the discipline of management and project management today.

During the early 1900s, two very popular mathematical project scheduling models were developed that are still widely used in practice by many organisations across the world. The first of which was the Program Evaluation and Review Technique (PERT), which was developed as part of the United States Navy and Lockheed Corporation Polaris Missile Submarine Program. The second was the Critical Path Method, which was developed to manage plant maintenance projects by DiPont Corporation and Remington Rand Corporation.

Since the 1950s, organisations have applied systematic project management tools and techniques to complex projects, and the 1950s was marked as the beginning of the modern project management era. Project management was recognised as a distinct discipline arising from the discipline of management (Cleland & Gareis, 2006:4).

The PMI was established in 1969 at the Georgia Institute of Technology as a non-profit professional organisation. The purpose of this organisation was to further establish the professionalism of project management by setting standards, conducting research and providing education on project management. The PMI first published the well-known Guide to the Project Management Body of Knowledge as a white paper in 1987, after which the first edition was published in 1996. PMBOK will be further discussed in Section 3.3.

PRINCE2 was first developed by the Central Computer and Telecommunications Agency (CCTA) as a UK Government standard for IT projects in 1989 (Richards, 2007:8), after which it was launched to the public for the first time in 1996. PRINCE2 will be further discussed in

Section 3.4.

From the history of project management, it is clear that project management is a discipline that manages, organises, plans and controls project elements in such a way as to meet a specific goal. In order to define “project management”, the term “project” must first be defined. Various definitions of the term “project” are listed in Table 3.1.

Table 3.1: Definitions of “project”

Source	Definition
Benjamin (2009:82)	“[A] project can be defined as a temporary endeavour consisting of a set of interrelated activities undertaken to create a unique product or service”
Boyer & Verma (2010:281)	“[A] project can be defined as a set of interrelated activities necessary to achieve established goals using a specified amount of time, budget, and resources”.
Chen, Jain & Tai (2006:206)	“[A] project can be defined as a temporary endeavour undertaken to create a unique product or service (i.e., in the present case, software) or another product by using software at a large scale”.
Conway (2000:7)	“[A] project can be defined as a temporary set of activities conducted to bring about a defined outcome”.
Gupta (2007:531)	“[A] project can be defined as undertaking a set of activities in a scientific and systematic way assuring efficiency in implementation to achieve defined goals in a fixed time”.
Haugan (2008:337)	“[A] project can be defined as a planned undertaking consisting of a process or set of procedures to accomplish a task”.
Holdsworth & Martin (1993:571)	“[A] project can be defined as: A complex set of interrelated activities which are undertaken to attain a predefined outcome”.
Knutson (2001:128)	“[F]rom a management perspective, a project can be defined as the work required to take an opportunity and convert it to an asset.
Muffatto (2006:168)	“[A] project can be defined as a set of activities characterized by defined objectives, limited resources and time constraints”.
Nagl (1996:21)	“[A] project can be defined as a short-term, limited project, as e.g. the development of a prototype system, or the development of revision 1 of a production system, elimination of errors and weaknesses of the system to get revision 2”.
Odum (2004:1)	“[A] project can be defined as a collection of both human and financial resources that are focused on the achievement of a specific set of goals and objectives”.
OGC (2009:3)	A project can be defined as “a temporary organisation that is created for the purpose of delivering one or more business products according to an agreed Business Case”.
PMBOK Guide (2008:5)	A project can be defined as “a temporary endeavour undertaken to create a unique product, service or result”.
Tiku (2002:1)	“[A] project can be defined as a scientifically evolved work plan devised to achieve a specific objective within a specified period of time”.
Tonchia & Cozzi (2008:3)	“[A] project can be defined as a set of complex, coordinated activities with a clearly defined objective that can be achieved through synergetic, coordinated effort within a given time, and with a predetermined amount of human and financial resources”.
Westland (2006:2)	A project can be defined as “a unique endeavour to produce a set of deliverables within clearly specified time, cost, and quality constraints”.
Westney (1992:35)	“[A] project can be defined as having many or few activities depending on the amount of detail in each activity”.

These definitions of “project” mention many of the same aspects/characteristics, which can be summarised in the following definition, which is the one adopted in this study:

A project is a unique and temporary undertaking to develop and deliver defined and changing project objectives and expectations within certain constraints.

Projects have certain characteristics that broaden the definition of “project”, as they:

- have a unique purpose and nature (Schwalbe, 2010:7; Westland, 2006:2);
- are temporary with a defined time frame (Schwalbe, 2010:7; Westland, 2006:2);
- have limited number of resources with limited availability, and requires it/them normally from different areas in the business (Schwalbe, 2010:7; Westland, 2006:2);
- change because there are certain levels of uncertainty in the project and business environment (Schwalbe, 2010:7);
- are designed using progressive elaboration (Schwalbe, 2010:7);
- should have a sponsor (Schwalbe, 2010:7);
- have an approved budget (Westland, 2006:2); and
- always have an element of risk (Westland, 2006:2).

After understanding the definition of the term “project”, the term “project management” can be defined. There are many project management definitions used in practice. The different definitions of “project management” are listed in Table 3.2.

Table 3.2: Definitions of “project management”

Source	Definition
Aliev, Fazlollahi & Aliev (2004:392)	Project management “can be defined as planning, directing, and controlling resources (people, equipment, material) to meet the technical, cost, and time constraints of the project”.
Blokdijk (2007:11)	Project management “is defined as the act of organizing and managing resources in a disciplined way so that a project at hand would be completed within the defined scope, quality, time and cost constraints”.
Bødker, Kensing & Simonsen (2004:27)	Project management “can be defined as establishing, maintaining, and overseeing the mutual commitments that make the project what it is”.
Bruegge & Dutoit (2010:578)	“[F]rom a functional point of view, project management can be defined as the process of guiding a project from the beginning through its complete duration to its end using a set of management models”.
DeCarlo (2004:32)	Project management “is the art and science of facilitating and managing the flow of thoughts, emotions, and interactions in a way that produces valued outcomes”.
Doss, Wallace & Webber (2003:88)	Project management “is ensuring that project operational actions adhere to the project parameters within the context of resources, time, and cost”.
Gottschalk (2006:47)	“[I]nformation systems project management is defined as the coordination and control of all of the activities required to complete an information systems project”.

Harrison & Lock (2004:6)	Project management “can be defined as the achievement of project objectives through people and involving the organisation, planning and control of resources assigned to the project”.
Hedeman & Vis van Heemst (2006:29)	Project management “can be defined as a co-ordinated organisation and management of a series of connected activities for delivering a pre-defined outcome”.
Hughes & Cotterell (2002:302)	Project management “is defined as the integration of project activity through the project life cycle to achieve the delivery of a defined product or service within prescribed constraints of time, budget, scope and quality”.
Kasse (2004:86)	Project management “can be defined as establishing and maintaining an environment that gets the work done”.
Kerzner (2004:2)	Project management “can be defined as the planning, scheduling, and controlling of a series of integrated tasks such that the objectives of the project are achieved successfully and in the best interest of the project’s stakeholders”.
OGC (2009:4)	Project management “is the planning, delegating, monitoring and control of all aspects of the project, and the motivation of those involved, to achieve the project objectives within the expected performance targets for time, cost, quality, scope, benefits and risks.”
PMBOK Guide (2008:6)	Project management can be defined as “the application of knowledge, skills, tools and techniques to project activities to meet the project requirements”.
Ribeiro (2009:9)	Project management “can be defined as the allocation, utilisation and tracking of resources to achieve a particular project objective within a specified period of time”.
Singh (2007:50)	Project management “can be defined as the planning, scheduling and controlling of a series of integrated tasks so that the goals of the project are achieved successfully in the best interests of the project’s stakeholders”.
Spinner (1997:4)	Project management “is defined as managing and directing time, material, personnel/labour, and costs to complete a project in an orderly, economical manner and to meet the established objectives of time, costs, and technical and/or service results”.
Thayer (1997:61)	Project management “is defined as a system of procedures, practices, technologies, and know-how that provides the planning, organizing, staffing, directing, and controlling necessary to successfully manage an engineering project”.
Westland (2006:2–3)	Project management can be defined as “the skills, tools and management processes required to undertake a project successfully”. It furthermore incorporates “a set of skills; a suite of tools; and a series of processes”.

After considering the origins of project management, what a project is, and looking at formulised definitions of project management and a project, project management can be summarised in the following definition, which is the one adopted in this study:

Project management is an approach followed with the intention of completing a project successfully. The approach can consist of several techniques, tools and processes of managing the project elements (project requirements and expectations, resources, stakeholders, etc.) to satisfy defined and changing quality, cost, scope and time constraints.

The three PMMs selected, PMBOK, PRINCE2 and APM, will be explained in the sections that follow.

3.3 Project Management Body of Knowledge

The Guide to the Project Management Body of Knowledge is an internationally recognised standard and project management guide, published by the PMI, which explains the fundamentals of project management for almost any type of project (software, construction, engineering, etc.). By 2010, there were more than 500 000 members and credential holders of the PMI in 185 countries, which demonstrates that this PMM is a well-known standard for the discipline of project management across the world.

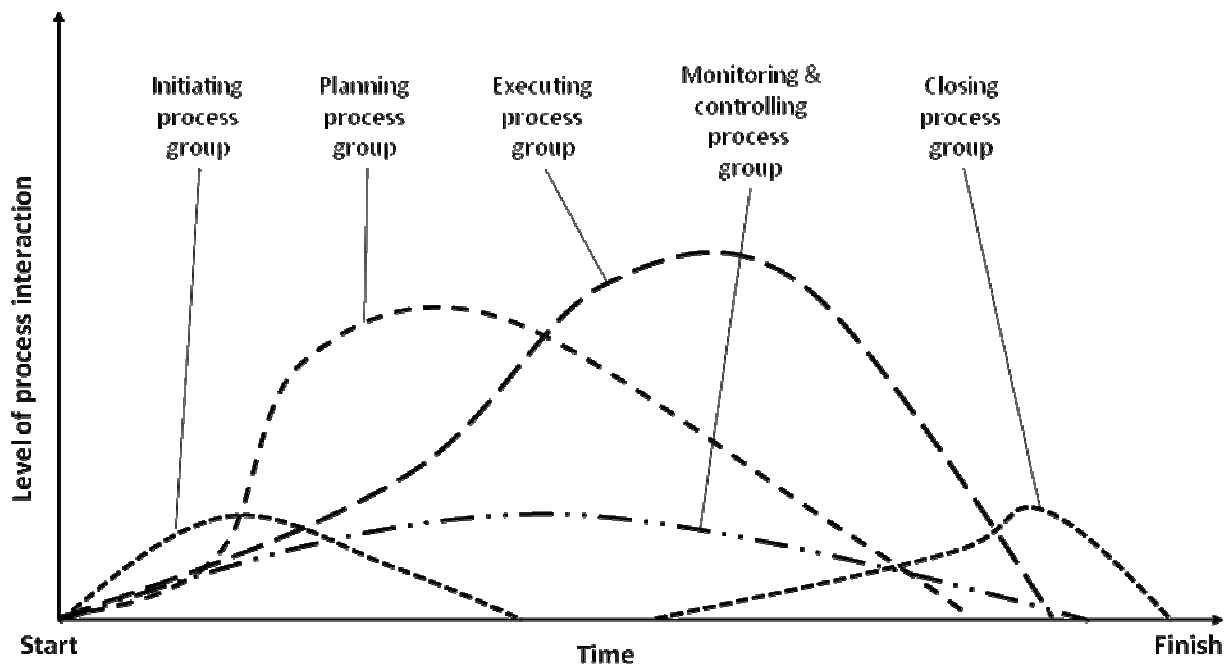
As presented in Figure 3.1, PMBOK was first developed and documented as a white paper in 1987, after which the first edition was released by the PMI in 1996. The second edition was released in 2000 and third edition in 2004. The fourth and latest edition was released in December 2008. There are not many substantive changes between PMBOK's third and fourth editions. Small changes have been made to make the document more consistent and user friendly (Snyder Stackpole, 2008). The PMBOK project management approach will be explained using mainly the fourth edition of the PMBOK Guide and the sixth edition of *Managing Information Technology Projects* (Schwalbe, 2010), which meets all the requirements of PMBOK's fourth edition.

PMBOK recognises 42 processes that are part of five main process groups and nine knowledge areas that are applicable to almost all projects. In this study, the five process groups and nine knowledge areas together with their associated processes will be explained.

3.3.1 PMBOK process groups

A process is a series of interrelated activities that must be completed in the correct sequence in order to reach a specific goal (Snyder Stackpole, 2010:2). Process groups interact with each other during all phases of the project (Snyder Stackpole, 2010:2; Ó Conchúir, 2010:24). There can be "different project phases, but all projects will include all five process groups" (Schwalbe, 2010:79).

The PMBOK project management process groups can be applied to each project phase or to a project as a whole. It is important to note that the process groups are not isolated events, and overlap all phases of a project. Figure 3.2 shows the process groups and the manner in which they overlap and relate with each other in terms of level of interaction and time of a project.



Source: PMBOK Guide (2008:41)

Figure 3.2: The level of process interaction and overlap of process groups over time

Each project management process group is characterised by certain tasks that must be completed. Each of these groups and tasks will be discussed in the sections that follow.

3.3.1.1 Initiation process

In order to initiate a project or a project phase, a project manager must be assigned to manage the project, the requirements of the business must be defined, and a sponsor for the project must be obtained (Snyder Stackpole, 2010:15). It is important to note that the initiation process is a part of all phases of the project, as before something can be planned or executed it must be initiated (Sherrer, 2009:54).

During this process, the project manager and team members are selected and organised, if they have not been selected during the pre-initiation phase. The pre-initiation phase is viewed by some organisations as a phase done before the project starts, during which the team members and the project manager are selected in order to ensure that the project can be initiated correctly and timeously.

The initiation process of a new project entails the completion of a business case and project charter. These documents specify the major scope, time, cost and quality goals and expectations of the project. Furthermore, they justify the projects, put major processes of understanding in place, and identify the main stakeholders for the project. After the completion

of the project charter, the stakeholders involved and affected by the project are identified.

3.3.1.2 Planning process

The planning process entails the development and maintenance of a workable scheme to ensure that the project satisfies and addresses the business's requirements. There are several plans and processes that can be executed during the planning process (PMBOK Guide, 2008: 48–55):

- project management plan development;
- collecting requirements;
- scope definition;
- work breakdown structure (WBS);
- activity definitions;
- activity sequence;
- activity resources estimation;
- activity durations estimation;
- schedule development;
- cost estimation;
- budget determination;
- quality planning;
- human resource plan;
- communications plan;
- risk management plan;
- risk identification;
- qualitative risk analysis;
- quantitative risk analysis;
- risk response planning; and
- procurement planning.

This process is extremely important for IT projects because it affects the project's cost, timeline and work effort to make changes if the project is not planned correctly. This is the reason that it is important to gather as many business and user expectations and requirements as possible before execution in order to ensure that the number of newly identified and changing project requirements are minimised. The planning process entails the completion of the project schedule, scope statement and cost estimation. Like the initiation process, the planning process can be executed in all phases of a project (Sherrer, 2009:55). The project team and manager must plan to define the work that must be done for the project. Some of the work to be done

includes the development of a time schedule that defines the activities that must be performed, a cost estimate of the cost of the project, as well as the number of resources that will be required.

3.3.1.3 Execution process

This process entails the organisation and coordination of resources to execute the various plans, actions and tasks to produce the deliverables, goals, products, services or results during the different phases of the project (Ó Conchúir, 2010:27; Sherrer, 2009:55). The execution process also takes place during all phases of a project after the planning process has been completed. Processes completed during the execution process include (PMBOK Guide, 2008:57–59):

- managing and directing project execution;
- performing quality assurance;
- acquiring the project team;
- developing the project team;
- managing the project team;
- distributing information;
- managing the expectations of stakeholders; and
- conducting procurement.

The process can be recognised by the completion of the actual work (tasks and activities) of the project to achieve the goals set out in the planning documentation. The most resources are normally required during this process.

3.3.1.4 Monitoring and controlling process

This process entails the continuous monitoring, measuring and controlling of project-related activities to ensure that the project team meets the objectives of the project. The project manager and team members continually monitor and measure the progress of the project against the project schedule and other plans in order to take corrective action if the project's scope, time, cost and quality constraints are threatened. The most common monitoring and controlling process is performance reporting, through which stakeholders (senior management, executives, etc.) can identify possible changes to be made to the project. Performance reporting ensures that senior management makes informed decisions based on facts instead of assumptions. Like all the process groups, the monitoring and controlling process takes place during each phase of the project and includes the following processes (PMBOK Guide, 2008:61–64):

- monitoring and controlling the project work;
- performing integrated change control;
- verifying project scope;
- controlling project scope;
- controlling project schedule;
- controlling project costs;
- performing quality control;
- reporting performance;
- monitoring and controlling risks; and
- administering procurement.

The ideal outcome of this process group is to complete the project successfully by satisfying all stakeholder scope, time, cost and quality expectations and constraints. Since change can occur during any phase of a project, this process group overlaps with all other process groups, which is the reason that the necessary change-control procedures must be put in place.

3.3.1.5 Closing process

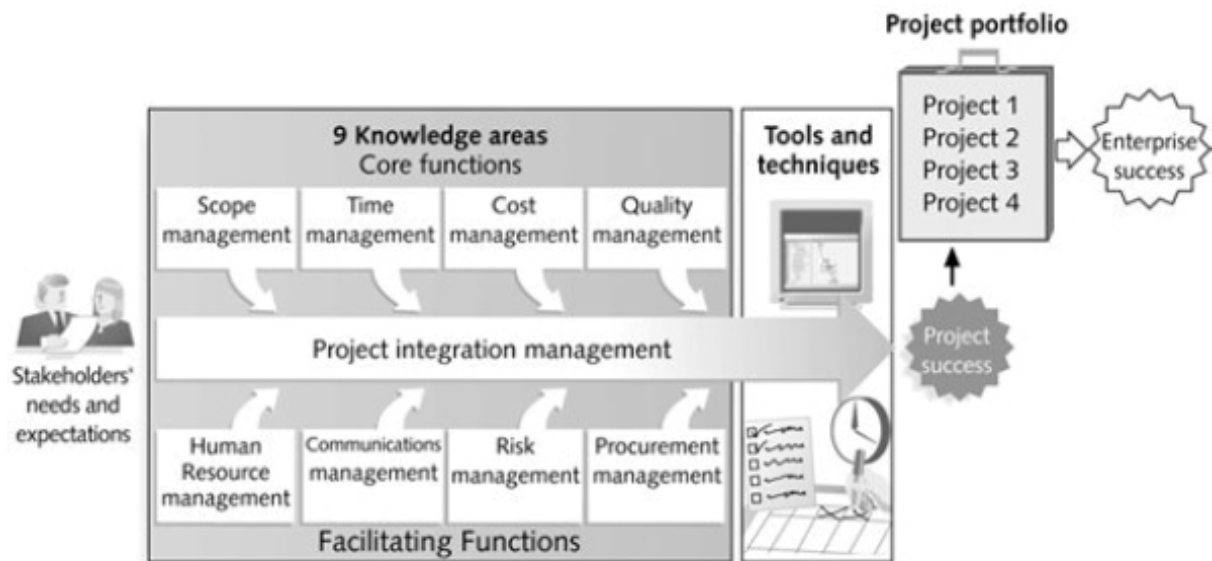
This process entails the formalised acceptance of the project's phases or the project as a whole and ending it efficiently and effectively (Ó Conchúir, 2010:28; Sherrer, 2009:56). Administrative, documentation and procurement activities are normally included in the closing process, such as completing a close-out report, documenting lessons learnt and completing the procurement. Like all other process groups, the closing process takes place during all phases of a project.

The process groups can be applied to each project phase or to a project as a whole. It is important to note that the process groups are not isolated events, and overlap all phases of a project.

3.3.2 PMBOK's nine knowledge areas

The nine knowledge areas of the PMBOK PMM can be best described for IT projects using the project management framework of Schwalbe (2010:10). In Figure 3.3, the project management framework consists of stakeholders' needs and expectations, nine knowledge areas (four core functions and four facilitating functions), tools and techniques. Stakeholders are the "people involved in or affected by project activities and include the project sponsor, project team, support staff, clients, users, suppliers and even opponents of the project" (Schwalbe, 2010:10). This figure clearly shows that if a project team utilises the nine knowledge areas, tools and techniques effectively to satisfy stakeholder expectations it will lead to a successful project,

which will eventually lead to a profitable organisation (enterprise success).



Source: Schwalbe (2010:10)

Figure 3.3: Project management framework

The nine project management knowledge areas explain the major abilities that must be executed by the project manager and team members, which are crucial to project success. In the figure, the nine knowledge areas have four core knowledge areas (scope, time, cost, quality) that leads to detailed project objects and constraints; four facilitating knowledge areas (human resources, communications, risk, procurement), which are the processes through which project goals are achieved; and integration management, which is an overarching function that integrates, affects and is affected by the nine knowledge areas. Table 3.3 below shows the manner in which the knowledge areas relate to the different process groups.

Table 3.3: Project management process group and knowledge area mapping

KNOWLEDGE AREAS	PROJECT MANAGEMENT PROCESS GROUPS				
	Initiating process group	Planning process group	Executing process group	Monitoring & controlling process group	Closing process group
Project integration management	- Develop project charter	- Develop project management plan	- Direct and manage project execution	- Monitor and control project work - Perform integrated change control	- Close project or phase
Project scope management		- Collect requirements - Define scope - Create WBS		- Scope verification - Scope control	

Project time management		- Define activities - Sequence activities - Activity resource estimation - Activity duration estimation - Develop schedule		- Schedule control	
Project cost management		- Estimate costs - Determine budget		- Control costs	
Project quality management		- Quality plan	- Perform quality assurance	- Perform quality control	
Project human resource management		- Develop human resource plan	- Acquire project team - Develop project team - Manage project team		
Project communications management	- Identify stakeholders	- Plan communications	- Information distribution - Manage stakeholder expectations	- Report performance	
Project risk management		- Risk management planning - Risk identification - Perform qualitative risk analysis - Perform quantitative risk analysis - Risk response planning		- Monitor and control risks	
Project procurement management		- Plan procurements	- Conduct procurements	- Administer procurements	- Close procurements

Source: PMBOK Guide (2008:43)

3.3.2.1 Project integration management

This knowledge area is critical for the satisfaction of stakeholder requirements. Project managers use integration management to assume responsibility for coordinating the team members, to focus on the project objectives and drive the project to successful completion, to make final decisions when there is conflict between team members and/or stakeholders, and to communicate key project information to senior management.

Project integration management ensures that all elements and processes of the different knowledge areas are integrated and well co-ordinated (Saladis & Kerzner, 2009:xiv; Sherrer, 2009:58) by managing and coordinating all the areas throughout the project life cycle. There are seven main processes involved in project integration management:

- *Developing project charter*: During this process, a project charter is compiled with the stakeholders to authorise the project formally and to list major cost, time, scope and quality constraints and requirements (Sherrer, 2009:68–69).

- *Developing project management plan:* During this process, a project management plan document is compiled that is coherent and that contain all the plans such as the project communication plan, procurement plan, risk plan, quality plan and project schedule (Sherrer, 2009:70).
- *Directing and managing project execution:* This process involves the execution of the activities, actions and plans within the project management plan to deliver on the business and user requirements. The outputs of this process are deliverables, requested changes, work performance information, project plan and project documentation updates (Schwalbe, 2008:130; Sherrer, 2009:75–76).
- *Monitoring and controlling the project work:* This process involves monitoring the progress of the tasks and actions of the project to meet the performance objectives of the IT projects and to manage and control possible changes. The outputs include requested changes, and project plan and documentation updates (Schwalbe, 2010:131; Sherrer, 2009:78–79).
- *Performing integrated change control:* This process involves the management, control and coordination of changes that may affect the project's deliverables and goals. The outputs of this process are updates on all change requests, the project management plan and other documentation related to the changes made (Schwalbe, 2010:131; Sherrer, 2009:81–82).
- *Closing project or phase:* This process entails the formalised finalisation of the project or a specific project phase. The outputs of this process are updates on organisational process assets and final result, service, or product transition (Schwalbe, 2010:131; Sherrer, 2009:85–86).

3.3.2.2 Project scope management

Defining the scope of a specific project is a very important part of the project because during this process deliverables that are included in the project are defined and controlled (Ó Conchúir, 2010:30). A deliverable can be described as a product that is produced as part of the project. The scope thus refers to all the work involved in creating the products by defining deliverables to be completed and the manner in which to control changes that occur (Schwalbe, 2010:178; Saladis & Kerzner, 2009:xiv; Sherrer, 2009:59). Scope management further ensures that the project manager, team and stakeholders have the same understanding of what objectives, services or products will be produced and what actions, tasks or processes will be followed by the project manager and team in order to produce them. Project scope management entails five main processes:

- *Collecting requirements:* During this process, the expectations and requirements to meet certain business objectives are gathered, defined and documented from the project's stakeholders. The outputs of this process are a requirements management plan, traceability matrix, and a stakeholder requirements plan (Schwalbe, 2010:179; Sherrer, 2009:102–

104).

- *Defining scope:* During this process, the tasks, actions and deliverables of the project are further defined and explained in order to develop a detailed definition of the product(s) and the project. The project charter is reviewed and possible change requests are approved. The outputs of this process are updated project documentation and the project scope statement (a project scope statement that explains clear and specific scope information, updated scope management plan and change requests; Schwalbe, 2010:179; Sherrer, 2009:107–110).
- *Creating WBS:* This process involves the creation of a WBS that divides and manages the deliverables of the project into small, manageable, and logical components. This makes it easier to delegate work to different team members, and for team members to realise what their responsibilities entail and the manner in which their responsibilities relate to the deliverables to be completed. Each team member also realises the influence his/her tasks will have on tasks assigned to other team members, if he/she does not deliver on the tasks assigned to him/her. The outputs of this process are a WBS dictionary, WBS, updated project documentation, and a scope base-line (Schwalbe, 2010:179; Sherrer, 2009:113–116).
- *Verifying the scope:* This process involves the final acceptance and signing off of the project's scope by the stakeholders. The acceptance includes all the deliverables and goals that must be reached within certain time, budget and quality constraints. If the project scope has not been accepted by either the stakeholders (client, sponsor, etc.) new recommendations or changes to the scope will be requested to be verified once again for acceptance. The main outputs of this process are requested changes and accepted deliverables (Schwalbe, 2010:179). Scope verification is very important to ensure that scope creep (project scope that extends as new requirements are identified and changes are requested) does not take place (Sherrer, 2009:118–119).
- *Controlling the scope:* This process involves the control of changes on the project scope. In practice, if users see what they can get, they always want more. The reason for this is that the client did not understand exactly what he/she required. Change to IT projects can have detrimental results if not managed and controlled correctly. The process also includes the identification, evaluation and implementation of approved changes requested as the project progresses (Sherrer, 2009:121). The main outputs are updates to the project management plan, documentation, requested changes, and work performance measurements (Schwalbe, 2010:179).

3.3.2.3 Project time management

Project time management is the process of estimating the time it will take to complete the

actions and tasks of the project deliverables according to stakeholder/business expectations and requirements (Sherrer, 2009:59). This process includes the development of a project schedule, containing sequenced tasks that show the time it will take to complete each task (Saladis & Kerzner, 2009:xiv), by whom each task is to be done, and the dependency (predecessors) on the other project tasks. The six main processes involved in project time management are the following:

- *Defining activities:* During this process, the activities, task and actions that must be executed by the project team and stakeholders to produce the different project and product deliverables are defined. The main outputs of this process are activity attributes, an activity list and a milestone list (Schwalbe, 2010:213). An activity list consists of a unique identifier, activity name and activity description in table format. Activity attributes provide more schedule-related information, such as predecessors, successors, legal relationships, leads and lags, resource requirements, constraints, imposed dates and assumptions related to the activity. A milestone list on the other hand does not have any duration and is viewed as a marker to identify that a series of activities has been completed. An activity has a certain duration, resource requirement, and cost, which is viewed as the smallest component of a WBS that must be executed together with other activities to achieve a certain milestone (Sherrer, 2009:131–133).
- *Sequencing activities:* During this process, the relationships amongst the defined activities are identified and documented. The main outputs of this process are updated documentation and project schedule network diagrams (Schwalbe, 2010:213; Sherrer, 2009:135).
- *Estimating activity resources:* This process involves the estimation of the number of resources (people, software, equipment, material, etc.) are required by the team to provide the deliverables of the project. The outputs of this process are a resource breakdown structure, activity resource requirements, and updates to project documentation (Schwalbe, 2010:213; Sherrer, 2009:141–142).
- *Estimating activity duration:* During this process, the project team estimates the time the project activities will take to complete in order to complete the project deliverables. The main outputs of this process are task duration estimates and updates to task attributes (Schwalbe, 2010:213; Sherrer, 2009:147–148).
- *Developing the schedule:* During this process, the project schedule is developed using the activity duration estimates, activity resource estimates and activity sequences. The outputs of this process include a scheduled base-line and data, project schedule, and updates to documentation (Schwalbe, 2010:213; Sherrer, 2009:154–157).
- *Controlling the schedule:* This process entails the management, control and monitoring of the project's progress, status and schedule. The outputs of this process are requested

changes, updates on organisational process assets, performance measurements and updates on the project plan and documentation (Schwalbe, 2010:214; Sherrer, 2009:178).

3.3.2.4 Project cost management

Project cost management is the preparation, estimation, control and management of a project's budget to ensure that the project is completed within the approved cost estimate (Sherrer, 2009:60). The budget must be realistic, containing accurate time and cost estimates to ensure that the defined project deliverables are successfully managed, controlled and completed (Saladis & Kerzner, 2009:xiv; Ó Conchúir, 2010:32). During cost management, the following processes are executed:

- *Estimating cost:* During this process, an estimate is developed of the cost of the resources (people, material, hardware, software, licences, etc.) for the sponsors in order to complete project deliverables successfully. The cost management plan should be created as part of the project management plan under integration management. The main outputs are the basis of estimates, activity cost estimates and updated documentation (Schwalbe, 2010:256; Sherrer, 2009:191).
- *Determining budget:* This process entails the allocation of project cost estimates to individual project activities or tasks over time to establish a base-line for measuring performance. The outputs of this process are funding requirements, a base-line of the cost performance and updates on documentation (Schwalbe, 2010:256; Sherrer, 2009:200).
- *Controlling costs:* During this process, changes to the project's budget are monitored and controlled by monitoring the project's status regularly. This process entails the monitoring of cost performance, informing stakeholders of approved cost changes that will affect the project, and ensuring that only appropriate cost changes are made to ensure successful deliverables completion. The outputs of this process are requested changes, forecast budgets and cost estimates, performance measurements, updates to process assets, and updates to the documentation and project plan (Schwalbe, 2010:256; Sherrer, 2009:205).

3.3.2.5 Project quality management

Project quality management ensures that the project will satisfy the defined needs and requirements of stakeholders and clients for which it was undertaken (Sherrer, 2009:60). By conforming to the needs of the clients, the project will be completed successfully, which is the reason that it must be viewed at the same level of importance as scope, time and cost management. There are three main processes involved in project quality management:

- *Quality planning:* During this process, the quality standards and requirements relevant to the project are identified and the manner in which to comply with or satisfy these standards and requirements is explained and documented. In IT, project quality standards can include

the response time for a technical issue regarding hardware or software consistency (without causing runtime errors, etc.). The outputs of this process are updated documentation, a quality management plan, quality checklists, quality metrics, and a process improvement plan (Schwalbe, 2010:295; Sherrer, 2009:231–232).

- *Performing quality assurance:* This process involves assuming the responsibility of assuring that quality standards are adhered to during the duration of the project by auditing the documented quality requirements and measurements thereof. Quality audits can be conducted, in which specific quality management activities are reviewed in a structured manner to identify lessons learnt that can improve current project performance or ensure better quality conformity in future IT projects. Furthermore, it involves the monitoring and periodical evaluation of the project's performance to ensure that the project satisfies the quality standards defined. The main outputs are requested changes, recommended corrective actions, updates to process assets, documentation, and the project plan (Schwalbe, 2010:295; Sherrer, 2009:243–244).
- *Performing quality control:* During this process, project deliverables and results are monitored to ensure that they comply with the relative quality standards, while identifying ways to improve overall quality in the project's life cycle. Quality audits can be conducted as with the quality assurance process above. The required changes are recommended, after the results of assessing the performance of executed quality requirements have been monitored. The outputs of this process include requested changes, validated changes and deliverables, quality control measures, updates to process assets, documentation, and the project plan (Schwalbe, 2010:296; Sherrer, 2009:247).

3.3.2.6 Project human resource management

Project human resource management involves the organisation, management and optimal utilisation of team members and stakeholders involved in the project (Sherrer, 2009:60). Human resource management entails the following processes:

- *Developing the human resource plan:* During this process, a staff management plan is developed that lists all project stakeholders' roles, required skills and responsibilities, including identified reporting relationships. The output of this process is a human resource plan (Schwalbe, 2010:343; Sherrer, 2009:271–271).
- *Acquiring the project team:* During this process, the required people are sourced and assigned to tasks of the project. Outputs of this process are staff assignments, resource calendars, and updates to the project plan (Schwalbe, 2010:343; Sherrer, 2009:279–280).
- *Developing the project team:* During this process, the project team's skill as a group and individually is enhanced to ensure that the team members perform work of quality throughout the project. In order to enhance the team's overall performance, the project

environment must be improved, which includes the competencies of the team and the way in which team members interact with one another. The outputs of this process are assessments on the team's performance and updates to environmental factors (Schwalbe, 2010:343; Sherrer, 2009:285).

- *Managing the project team:* The management of the project team entails the project manager ensuring that the team members execute their assigned tasks correctly and timeously; providing feedback timeously; resolving problems, risk, issues and conflict; tracking team member performance, motivating team members; and managing changes to the project in order to enhance the project's performance and progress. The outputs of this process are requested changes, and updates to the project plan, organisational process asset, and environmental factors (Schwalbe, 2010:343; Sherrer, 2009:299).

3.3.2.7 Project communications management

Project communications management ensures that accurate project information is generated, collected, disseminated, stored and distributed to different respondents (Sherrer, 2009:61). There are mainly five processes in project communications management:

- *Identifying stakeholders:* During this process, everyone involved in or affected by the project is identified and their interests and impact on the project are documented. The outputs of this process are a stakeholder management strategy and register (Schwalbe, 2010:383; Sherrer, 2009:220).
- *Planning communications:* This process entails determining the communication and information requirements (who, when and how) of the stakeholders and defining a communication approach. The outputs are a communication plan and updated project documentation (Schwalbe, 2010:384; Sherrer, 2009:227).
- *Distributing information:* This process involves making the information required by stakeholders available in a timeous manner. The output of this process is updates to organisational process assets (Schwalbe, 2010:384; Sherrer, 2009:332).
- *Managing stakeholder expectations:* This process involves the management of stakeholder communications to satisfy the expectations and requirements of stakeholders and to find solutions for problems, issues, risks and conflict situations they might have. The outputs of this process are requested changes, updates to process assets, documentation and the project plan (Schwalbe, 2010:384; Sherrer, 2009:335).
- *Reporting performance:* This process involves the collection, analysis and distribution of performance information, which includes status and progress reports, progress measurements, milestone or deliverable reports, and forecasting. The outputs are requested changes, performance reports, and updated process assets (Schwalbe, 2010:384; Sherrer, 2009:338).

3.3.2.8 Project risk management

A project risk can be defined as “an uncertainty that can have a negative or positive effect on meeting project objectives” (Schwalbe, 2010:425). A risk is a possible threat to or advantage of the project that is foreseen and that must be managed. Project risk management includes identification, analysis and preventative actions and responses to project risks (Sherrer, 2009:61). In project risk management, there are six main processes involved:

- *Risk management planning*: During this process, a decision is made of the manner in which to plan and approach the risk management activities for the project. Risk management activities can be discussed and analysed for a project by reviewing the project scope statement, project management plan, environmental factors, and organisational process assets. The output of the risk planning process is a risk management plan (Schwalbe, 2010:427; Sherrer, 2009:356).
- *Identifying risk*: During this process, the risk likely to affect the project is determined and the characteristics of each are documented. Risks cannot be managed if they have not been identified, which is the reason that it is important to identify potential risks as early as possible and during the progress of the project to ensure that all risks and changes in the project are identified. The output of this process is the start of a risk log or register (Schwalbe, 2010:427; Sherrer, 2009:365).
- *Performing qualitative risk analysis*: During this process, the identified risks are prioritised based on their occurrence probability and impact using various tools and techniques that exist in the marketplace today. The outputs of this process are updates to the risk register (Schwalbe, 2010:427; Snyder Stackpole, 2010:105; Sherrer, 2009:376).
- *Performing quantitative risk analysis*: This process can be done together with qualitative risk analysis or separately. It includes the numeric estimation of the effects that risk may have on project deliverables and activities. This analysis process entails data collection, quantitative risk analysis, and modelling techniques. The output of quantitative risk analysis is an updated risk register (Schwalbe, 2010:427; Sherrer, 2009:381).
- *Risk response planning*: During this process, the necessary steps are taken to enhance opportunities or positive risks and reduce threats or negative risks by creating options and actions in order to meet project deliverables and objectives. This process entails the development of a response strategy to negative and positive risks. The basic response strategy to negative risks includes risk avoidance, risk acceptance, risk transference and risk mitigation, while the response strategy to positive risk includes risk expectation, risk sharing, risk enhancement and risk acceptance. The outputs are results to update the risk register, risk-related contractual agents, documentation, and the project plan. (Sherrer, 2009:391).
- *Monitoring and controlling risks*: During this process, the risk management processes are

executed, in which identified and residual risks are monitored, new risks are identified, risk response plans are carried out, and the effectiveness of the risk strategies is evaluated through the project's life cycle. The main outputs include recommended corrective and preventative actions, requested changes, and updates to the risk register, project plan, and process assets (Schwalbe, 2010:428; Sherrer, 2009:398).

3.3.2.9 Project procurement management

Project procurement management includes the acquisition or procurement of services, resources, products and goods for the project using external organisations or sources (Saladis & Kerzner, 2009:xiv; Sherrer, 2009:62). The six processes for project procurement management are:

- *Planning procurements*: This process involves the determination of how, where and what to procure. During the planning process, purchasing decisions must be made of what should be outsourced, what types of contracts must be put into place, and the work, including the approach, must be explained to potential suppliers and contractors. Outputs of this process are requested changes, procurement plan, statement of work, make-or-buy decisions, source selection criteria, and procurement documentation (Schwalbe, 2010:465; Sherrer, 2009:416).
- *Conducting procurements*: During this process, the seller responses are collected, the seller is selected and the contract is awarded to the most appropriate service provider or contractor. Outputs of this process include requested changes, selected sellers, awarded procurement contracts, resource calendars, and updated documentation and project plan (Schwalbe, 2010:465; Sherrer, 2009:428).
- *Administering procurements*: During this process, the relationship with the contractors and service providers is managed, the performance of awarded contracts is monitored, and the necessary changes are made. The outputs of this process are requested changes, procurement documentation, and updates to process assets and the project plan (Schwalbe, 2010:465–466; Sherrer, 2009:436).
- *Closing procurements*: This process ensures the settlement and completion of all contracts of selected service providers and contractors, including the resolution of any open issues, problems and items. The outputs of this process are updated process assets, closed contracts and procurements (Schwalbe, 2010:466; Sherrer, 2009:442).

After discussing the five process groups and nine knowledge areas of PMBOK, PMBOK will be evaluated by looking at its area of application, advantages, disadvantages, and integration with other methodologies.

3.3.3 Evaluation of PMBOK

PMBOK is a globally accepted PMM and is considered to be the global standard for managing projects in general. Although accepted as a globalised standard, there is continual improvement taking place with the different editions having been developed since 1969, which demonstrates that there is room for PMBOK enhancement. With each release, the developers have admitted that the new release attempts to address flaws in the previous edition to make the PMM even better. Nothing is perfect, including PMBOK – there is always room for improvement and there will always be identified flaws. Not every project delivered successfully or unsuccessfully is published by organisations, which makes it difficult to validate the true value and effectiveness of PMBOK in practice.

Application area

PMBOK was found to be successful in Asia for large-scale IT projects, where the Malaysian government used PMBOK to develop a Multimedia Super Corridor Integration Architecture framework for large-scale government applications (Phin, Zamani & Kheng, 2005:1–8). American private-sector retail organisations and governmental authorities (Transport Authority) have embraced PMBOK as a guide to assisting in delivering projects successfully (Crawford & Blackburn, 1996:2). The correct application of PMBOK concepts was found to improve the successful delivery of complex enterprise resource planning systems (Diaz, 2006:6).

PMBOK is the oldest and the most established PMM with the original design intent of being applied to any type of project, whether it is a construction, engineering, or an IT project. This makes PMBOK a generalised PMM that can be utilised by all industries in the marketplace. PMBOK was thus not specifically designed with the intent of managing IT projects, as is the case with PRINCE2. PMBOK can be applied to large, small and complex projects with any shape and size – taking into account that it might become too administratively intensive for a project manager managing a small to medium-sized IT project alone. The project manager must be on site with the developers, architects and stakeholders. He/she must not make the mistake of focusing only on adhering to the PMBOK processes and continuously submitting reports, instead of ensuring that the actual work is completed. It is better to miss a process in PMBOK than to miss a deliverable that could result in the failure of a project.

Advantages

According to Saladis and Kerzner (2009:vii), PMBOK is a guide that provides guidelines to organisations on “how to manage projects, irrespective of the characteristics”, and it is also used as the foundation to create other PMMs. Webber and Webber (2009:18–21) explain some advantages that PMBOK provides:

- supports the use of earned value to detect problems and monitor progress;
- exhibits its governmental background by its procurement management chapter;
- views people as very important by team acquisition planning, work recognition, training, and team building;
- explains a responsible project manager that interacts with a project sponsor, while the PRINCE2 project manager is given strategic direction from the project board on the manner in which to manage daily issues.

Furthermore, PMBOK has certain strengths, which provide some considerations of value (Van Bon & Verheijen, 2006:215):

- It is recognised as an international standard in the profession of project management.
- There are many different templates and other documentation that are associated with PMBOK that can be used in different applications and areas. PMBOK can be applied to any type of project because it is generic.
- Worldwide different organisations and industry sectors participate extensively in project management.
- It is appropriate for executing and delivering successful IT projects.
- There are internationally recognised certification programmes, such as Project Management Professionals (PMP) and Certified Associates in Project Management (CAPM), to guarantee accredited and skilful project managers.
- There is continuous improvement and evolution in accordance with the most up-to-date concepts of quality management.
- Similar to other frameworks such as ITIL, ISO, and COBIT, it focuses on process.

From the perspective of PMBOK's knowledge areas, the advantage above that of other PMMs is that it covers procurement and human resource management, which are very important management areas of any project. The integration management function of PMBOK ensures that all the knowledge areas merge to ensure that each part of the project is managed as a whole and not as individual management areas. PMBOK addresses integrated change control that is embedded within integration management, which is critical for the delivery of a project in a constantly changing environment. PMBOK explicitly and adequately covers the most important components of any project – cost and time management – as knowledge areas better than other PMMs. The way in which PMBOK manages risk is more effective and more comprehensive than other PMMs applied in practice. Furthermore, PMBOK is the only PMM that has quality assurance and control processes to manage and control the delivery of valuable products to clients actively.

The PMBOK process groups are well integrated into the various knowledge areas. Furthermore, PMBOK emphasises a monitor and controlling function across all knowledge areas, which is crucial for the successful delivery of IT projects within a continual changing environment. Another benefit is that PMBOK encourages a formalised close-out in order to ensure that what was promised in the original business case and project charter was actually delivered to client specifications.

Disadvantages

There are however some shortcomings, deficiencies and disadvantages of PMBOK. Phin *et al.* (2005:7) emphasise that the use of PMBOK “does not guarantee that your project will be smooth sailing, as there are various external factors that may influence the success of a project”, of which people is the main factor because they are responsible for the planning and execution of the project. Siegelau (2006:7) implies that a project manager using PMBOK might struggle to pull the different components of PMBOK together when initiating a project, and suggests that PRINCE2 can be used to assist with project initiation.

With regard to the life cycle, Koskela and Howell (2001:1) emphasise that the unclear relationship between planning and execution causes execution not to be managed systematically, and that the role of planning is not clearly defined, resulting in short-term planning that is carried out poorly or not at all. Furthermore, the function of control is not seen as a learning process, but rather as a function of taking corrective action and measuring performance and results.

Globerson and Zwikael (2002:58) found that communications management and risk management are the knowledge areas in which the poorest quality planning is done. The poor quality of work completed in these areas is the result of a lack of communication and risk management tools and techniques, which will only be resolved when training and tools are provided to manage and resolve risks and communication with stakeholders.

PMBOK further also struggles to deliver on E-Government project challenges where Furlong and Al-Karaghoul (2009:1) explains that the initiation phase is “identified as being particularly weak and inadequate in addressing e-Government initiatives and requirements”. Some weaknesses of PMBOK that could be improved are (Van Bon & Verheijen, 2006:215):

- the lack of tools applied in practices as real-life examples; and
- the lack of aspects and components required from the real-world implementations.

PMBOK is a governance framework that assists project managers in ensuring that they address

different critical management areas to deliver IT projects successfully. It is not simple or easy to use the framework, as the project manager must undergo intensive training to utilise the PMM correctly. He/she also runs the risk of being blamed for a project's failure because PMBOK was applied incorrectly, although there may have been many other factors that may have caused the failure. The main responsibility of a project manager is to ensure that the work is done correctly within time, cost, quality and scope constraints. If a framework of endless processes hinders a project manager from doing his/her work, this indicates that it may not be worthwhile to seek to adhere to a PMM if it hinders the project manager from doing the actual work. For this reason, a PMM must be simple, easy to use, and assist the project manager in completing the actual work, instead of being an administrative burden.

Concerning PMBOK in its current form of a management framework, some important management areas are missing, such as documentation management and issue management. Documentation management is an integral part of any organisation and version control across different IT projects is of utmost importance to ensure that the team and stakeholders reference the same requirements or piece of work completed. Much confusion can be created if not everyone is up to date on what the latest versions are, especially on large IT projects where different divisions use the same documentation to make decisions. Furthermore, PMBOK addresses risks (something foreseen to have a positive or negative impact on the project), but it does not clearly state the manner in which issues (something that already happened that already has a negative impact on the project) should be addressed. When managing issues, the milestone or deliverable affected must be listed and a resolution action and person responsible must be identified and executed to ensure that the issue does not affect the project further. Many IT projects fail because issues are not addressed and resolved quickly.

The PMBOK framework is too structured and cannot change or adapt quickly to its immediate environment, which raises a concern because an IT project has the tendency to change constantly because of factors that were not taken into account when planning was done. For example:

A sponsor for a project have not been assigned yet and senior management gave the instruction to a group of people across different divisions of an organisation to deliver a project. Furthermore, the individuals assigned to the project were not given the mandate to deliver the project, as senior management was simply too busy. Although no sponsor has been assigned to the project and although no mandate is in place to execute a project, the successful completion of the project is added to the assigned individuals' performance contract, requiring that they deliver. Failure to deliver might cause a person to lose his/her contract.

In a business environment like the one explained, a structured PMM will not function well. In the midst of chaos, especially in a rapidly growing organisation, PMBOK will suffer. The organisation needs something simple, something that can adapt quickly to its immediate needs – not something that requires intensive training only to understand the functioning of the PMM.

Often, IT projects are initiated only with a project manager as the only team member, which is in conflict with the initiation phase during which team members must be selected and acquired before planning is commenced. Sometimes even planning is done only by the project manager, as the team members have not been bought onto the project because it sometimes takes months to find the right people within or outside an organisation, while project timelines must be met. Planning is then done without having the right team members to provide input in order to determine an executable project schedule. This is a significant risk; the project team must be part of planning, as the project schedule is based on the individual experience and expert knowledge of each team member. For example, the project might require the expert advice of a systems architect to ensure that the estimated timelines are correct. If this is done by a project manager, the time estimates will be incorrect, and this may cause the failure of the project. Furthermore, planning can only be done to a certain point because if it is not based on facts – it is not worth much, and then if the facts change – then there might be a problem if a structured approach is followed. PMBOK has much to offer, but it is designed for an ideal world of project management, while the real world is not ideal, IT projects change on a daily basis and there is a need for a PMM that does the same. Structured PMMs also result in a large administrative burden, which hinders the project's progress – because everyone has to adhere to the PMM provided.

PMBOK does not often make mention of the use of a feasibility study, which is a concern because for most IT projects there might be more than one solution for satisfying client expectations within different time, cost and quality constraints. A feasibility study is necessary for determining the advantages and disadvantages of each proposed solution and thereby selecting a solution that would satisfy customer expectations.

There is a lack of organisational change management, which means that the business must be informed that a new project is in development and be kept up to date on the progress and roll out of the project. By keeping the business informed of what will happen and what benefits the new system will bring, the business users will become used to the idea. In many cases, there is significant resistance to change, which could have been prevented if the business users have been kept informed of the project's progress. Most people, especially elderly people, are scared of losing their jobs when a system is implemented, which is why they will resist it to ensure an income for them and their families.

Integration with other methodologies

Rico *et al.* (2009:47) explain that the self-contained series of iterations executed by ASDMs cover all the project management practices found in PMBOK. Many practitioners state that PMBOK supports ASDMs and that their combination would benefit the successful delivery of IT projects (Sliger & Broderick, 2008:xvii; Fitsilis, 2008:381; Perry, 2009:494). The combination of PMMs and ASDMs will be reported on in Chapter 4.

Furthermore, PMBOK has the ability to be integrated with the Rational Unified Process (RUP) to create an incremental, iterative and integrated RUP–PMBOK project management framework that is used for the acquisition and installation of software packages, new development project, application conversion projects, and technical environment migration projects (Bainey, 2004:394–399). PMBOK also has the ability to be integrated with other frameworks such as CMMI (Selig, 2008:42; Von Wangenheim, Da Silva, Buglione, Scheidt & Prikladnicki, 2010:749–757). Integration with other SDMs, such as RUP, and other methodologies emphasises that PMBOK and other SDMs alone are not sufficient to deliver IT projects. For this reason, this study sought to determine whether PMBOK could be combined with the best practices of other PMMs and ASDMs to deliver IT projects in a constantly changing project and business environment.

Because PMBOK is a worldwide standard for project management and viewed by practitioners as the benchmarking framework for project management, incorporation of some of the aspects of PMBOK will strengthen any other methodology when a project needs to be delivered successfully.

3.4 Project IN Controlled Environment

PRINCE was first developed by the CCTA as a UK Government standard for IT projects in 1989 (Richards, 2007:8). This PMM was initially based on a PMM called PROMPT, which was developed in 1975 by a organisation called Simpact Systems Ltd. The CCTA adopted PROMPT in 1979, which was superseded by PRINCE for all UK Government projects.

It is a structured, scalable, flexible and adaptable PMM for “effective project management” that covers the management, organisation, and control of a project (Richards, 2007:8). PRINCE2 provides (Bentley, 2005:5):

- terms of its business investment and return on investment, a controlled change-management approach;
- continuous and active user involvement through development to deliver a final product that meets environmental, service, functional, and management requirements; and

- effective control of resources doing development.

PRINCE2 adopts the principles of good project management to avoid possible project failure, which are:

- projects are finite, meaning that each has a specific and controlled start and end;
- projects must be managed in order to be delivered successfully;
- stakeholders in a project have to understand the reason the project is required, what needs to be achieved, the manner in which the defined objectives are to be achieved and what their accountabilities and responsibilities are.

PRINCE2 is a PMM that has been applied by professionals (business and public) across the world. By 2007 an estimated 250 000 people had completed the PRINCE2 practitioners' examination and this figure has been growing year on year by 20% (OGC, 2009:foreword). It has become a community, comprising (Murray, 2007:2):

- the UK Government (it is owned by the OGC);
- more than 120 accredited training organisations, providing training around the globe in seventeen languages;
- a documented PMM;
- around fifteen accredited consulting organisations;
- an accreditation body (APM Group Ltd);
- software tools (in the last PM Software Tool source book there existed fifty-two tools supporting PRINCE2);
- an official user group (The Best Practice User Group) and several others in many countries;
- more Internet pages that reference PRINCE2, than any other PMM; and
- several online discussion forums dedicated to PRINCE2.

PRINCE2 provides a PMM that is divided into manageable units that enable the effective control of resources and progress monitoring throughout the life cycle of the project. PRINCE2 provides projects with (Richards, 2007:8):

- regular reviews of progress with regard to the business case and project plan;
- flexible decision points;
- deviation from the plan is managed and controlled automatically;
- timeous involvement of management and stakeholders during the project;
- excellent communication channels between the organisation and the project team; and
- concurrence on the quality and continuous quality monitoring of deliverables.

The business case of the project plays a very important part of the PRINCE2 PMM. It justifies

the deliverables of the project and it is important to regularly check whether the project is still satisfying the deliverables as set out in the business case. Furthermore, PRINCE2 focuses on the products or results that are “produced by the project and not the activities to produce them” (Bentley, 2005:5). It is very important for the project manager to deliver the project within time, cost, scope and quality constraints as stipulated in the business case and project plan. Project managers using PRINCE2 will be able to (Richards, 2007:8):

- create terms of reference before the project is commenced;
- utilise a defined structure for continues communication and delegation;
- do accurate planning by dividing the project into manageable stages;
- ensure that management provide committed resources when approval is received to proceed with the project;
- keep management and stakeholders meetings to a minimum and discuss only what is relevant to the project; and
- provide concise and regular management reports.

PRINCE2 launched a new edition that has significant changes on 16 June 2009, although it retained the basic principles of the 2005 edition. The re-launch of PRINCE2 was the result of a long consultation process, which started in November 2006. The main changes between the 2005 and 2009 edition of PRINCE2 are that there are two guides for PRINCE2 (Murray, 2009:3):

1. *Managing Successful Projects with PRINCE2 (ver. 5; OGC, 2009):*

This guide has been designed for project managers, project support, and the members of the project team to enable them to deliver their IT projects successfully by using the principles, processes and techniques explained. The current study mainly considers the first guide. Furthermore, the guide explains the manner in which PRINCE2 can be applied in difficult project environments. The guide provides the project manager and team with the following guidance (Murray, 2009:3):

- what is expected of each of them
- what has to be done if things go wrong;
- what expected decisions might need to be made;
- what information is needed and has to be supplied;
- where direction and support can be provided; and
- the manner in which to tailor PRINCE2 to suit the project at hand.

2. *Directing Successful Projects with PRINCE2 (ver. 1):*

This guide is written for executive managers involved for the first time in a project to explain what is expected of them and what is required of the project board or executive project

committee and the project manager. Not only does the guide provide an overview of the PRINCE2 method, it also explains agendas, checklists and reviews for the executive project committee. The guide provides guidance by explaining (Murray, 2009:4):

- what is expected of the executive project committee;
- what should be expected of the project manager;
- determining whether the project manager is applying PRINCE2 correctly and appropriately;
- the manner in which to delegate authority to project managers and maintain control;
- the type of decisions expected to be made, and the information available/required to provide assistance during the decision-making process;
- the manner in which PRINCE2 can be tailored for different types of projects; and
- the composition of a project board or executive project committee.

The 2009 edition of PRINCE2 has retained its core characteristics of being a PMM that is applicable to any type, scale, complexity or culture of a project. The areas of comparison between the 2005 and 2009 editions of PRINCE2 are presented in Table 3.4.

Table 3.4: Comparison between PRINCE2 2005 and PRINCE2 2009

AREA	PRINCE 2005	PRINCE 2009
Principles	None	7 principles
Themes/components	<i>8 components:</i> • Business case • Organisation • Plans • Controls • Management of risk • Quality in a project environment • Configuration management • Change control	<i>7 themes:</i> • Business case • Organisation • Quality • Plans • Risk • Changes • Progress
Processes	<i>8 processes:</i> • Starting up a project • Directing a project • Initiating a project • Controlling a stage • Managing a stage boundary • Closing a project • Planning	<i>7 processes:</i> • Starting up a project • Directing a project • Initiating a project • Controlling a stage • Managing a stage boundary • Closing a project
Sub-processes	45 codified sub-processes comprising prescriptive actions.	40 activities comprising recommended actions. No codes.
Techniques	3 techniques explained (product-based planning, change control, quality review).	2 techniques explained (product-based planning and quality review) and numerous cross-references to techniques from other bodies of knowledge, including “soft” aspects.
Project environment	Not covered.	Context rich guidance on tailoring the method according to the project’s environment, including: • Projects in a programme • Commercial client/supplier relationships

		• Multi-owned projects • Alignment with other life-cycle models and bodies of knowledge • Project scale
Management products	36 management products	26 management products with explicit guidance on their evolution and which ones can be combined.
Roles	10 roles (project board, senior user, executive, senior supplier, project manager, team manager, project assurance, project support, configuration librarian, project support office).	8 roles (project board, senior user, executive, senior supplier, project manager, team manager, project assurance, project support). Suggested competences for each role. Project board roles now include duties and behaviours.
Checklists	Component-based checklists.	Process-based checklists. Governance checklist aligned to APM's governance principles.
Project board guidance	Embedded within the guide and aimed at project managers rather than project board members.	Role specific guidance for senior managers who sponsor or direct projects, including: • What makes a good project board • Suggested agendas for project board reviews • Checklist of key decisions for each project-board review • Pre- and post-project responsibilities

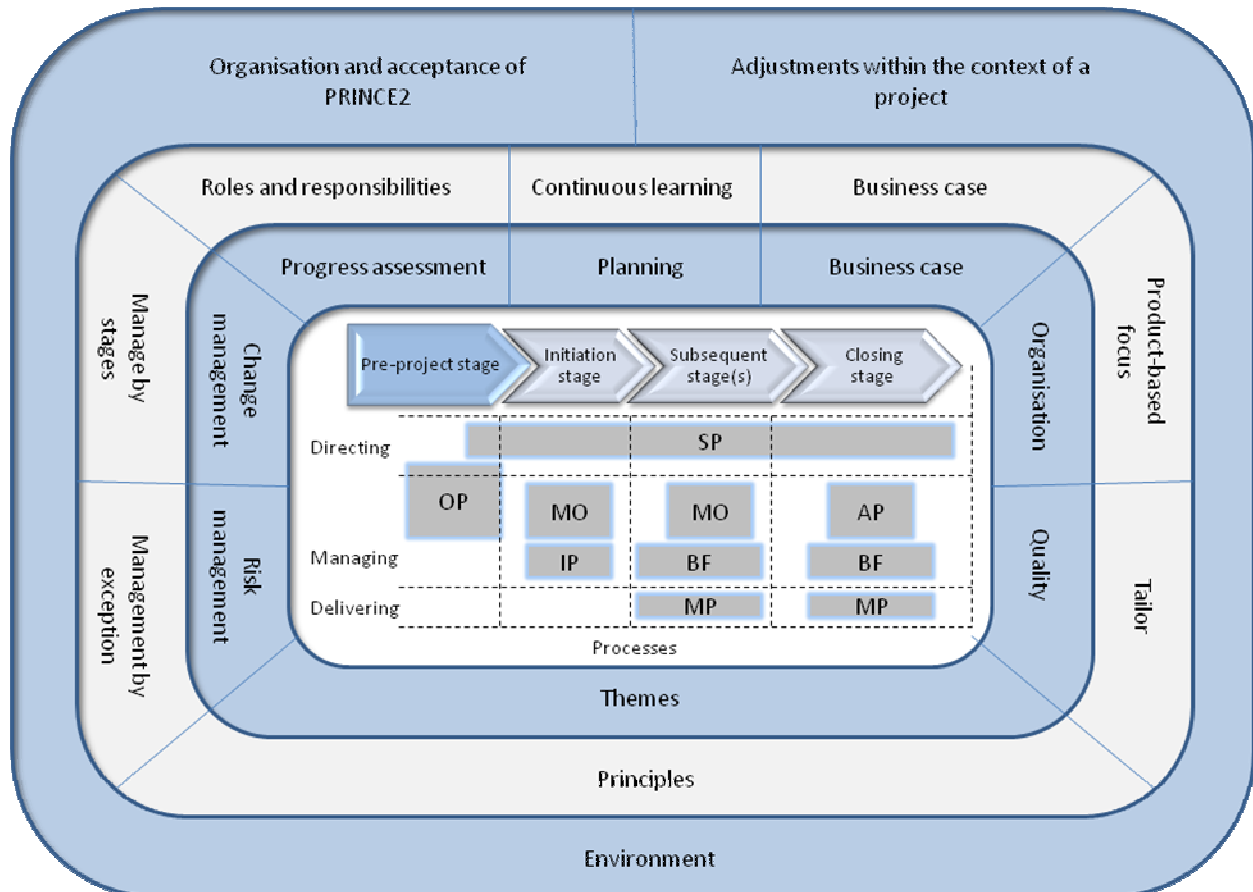
Source: Adapted from Murray (2009:5–6)

From the table above, it is clear that certain terminology has changed in order to place PRINCE2 projects in a business context rather than an academic context. The key improvements of PRINCE2 2009 are (Murray, 2009:4):

- It is less prescriptive and more flexible.
- It is less theoretical and more practical – based on the input of over 170 organisations and validated through pilot studies.
- It now has a set of clearly defined principles. These principles can be used as a check that PRINCE2 is being applied in the spirit in which the methodology has been designed – not too rigidly nor superficially.
- The need to tailor the methodology is explicitly stated and guidance on the manner in which to do so is provided.
- It has been designed to align with other OGC products: Managing Successful Programs (MSP), Management of Risk (M_o_R) and the new Portfolio, Programme, and Project Office (P3O) guide, which will enable users to integrate all four methodologies and frameworks seamlessly.
- The linkage with other standards and bodies of knowledge is clearly shown.
- The importance of the soft aspects of project management is emphasised.
- PRINCE2 is not bureaucratic – the method requires information and decisions, not excessive documents and meetings.

3.4.1 PRINCE2 structure

As mentioned above, in this study PRINCE2 is discussed with reference to OGC (2009) and Portman (2009). There are very limited resources that explain the use and effectiveness of PRINCE2 2009. For this reason, only pre-2009 literature will be discussed with reference to the different areas of PRINCE2 2009, as much terminology within the areas stayed the same as that of PRINCE2 2005. The structure of PRINCE2 2009 is summarised in Figure 3.4.



Source: Portman (2009:2)

Figure 3.4: The structure of PRINCE2

In the figure above, PRINCE2 is divided into the pre-project stage, initiation stage, continuation stage (subsequent stages) and closing stage. From the perspective of a project board and project manager, these four stages distinguish amongst seven main processes (Portman, 2009:3), which will be explained later in the chapter. During the pre-project stage, someone in the organisation comes up with an idea that might bring value to the organisation. The idea can be triggered by a business objective or organisational strategy, or anything else and is called the project mandate in PRINCE2 terms. In the initiation phase, a decision is made regarding whether to pursue the idea. During this process, funding must be obtained, project management strategies and controls are defined, detailed planning is done, and a business case is developed (OGC, 2009:114). In the subsequent stages, work is completed by the project team on a stage-by-stage basis. The project manager delegates work to the team and tracks the

progress of the project to ensure that it corresponds with the original project plan in order to provide feedback to the project board (OGC, 2009:114). In the final delivery stage, the project is decommissioned to ensure that the client has the applicable knowledge to use and own the system delivered. Once the successful operational hand-over has been completed and the project board is satisfied, the project can be closed out and signed off. The PRINCE2 structure will be explained from the outside (environment) to the inside (processes) of the above figure.

3.4.1.1 PRINCE2 project environment

PRINCE2 has the ability to adapt to different types of projects and to different types of project environments in order to ensure that IT projects are completed successfully. Because of the fast-paced, continually changing and evolving business environment today, it is important to ensure that an organisation and its PMM embrace change and adapt to ensure survival in the competitive marketplace. During the organisation and acceptance of PRINCE2, attention is drawn to the following aspects (Portman, 2009:8):

- the project management process must be owned and understood by the project manager and team members;
- standardised definitions and documentation templates;
- development strategy and training;
- guarantees;
- guidelines and rules for PRINCE2's scalability;
- business processes integration; and
- tools.

When an organisation accepts the PRINCE2 PMM, it will have the ability to deliver different types of IT projects in a dynamic environment. The project manager and project team can be guided by the following aspects to adapt to the unique circumstances of the project (Portman, 2009:8):

- “the size of the project;
- the independence of the project (or is it a sub-division of a program?);
- the type of project (product development, policy, construction, etc.); and
- use of specific methods or frameworks in the project.”

During the adoption process, the project team will need to agree on the following (Portman, 2009:8):

- allocated roles and responsibilities;
- scope, nature and number of project stages;
- project management products to be utilised;

- tolerance levels;
- utilisation of PRINCE2 processes; and
- reporting requirements and assessments to be completed.

3.4.1.2 PRINCE2 principles

The seven PRINCE2 principles guide project managers, the project team and executive managers in delivering IT projects successfully within a unique project environment. The approach to be taken in order to apply the seven PRINCE2 principles is explained in the seven themes below (Section 3.4.2.3). The seven principles are (OGC, 2009:11–14; Portman, 2009:5):

1. **Continuous business justification:** Business justification is ensured by applying the *business case* theme, according to which the project team continually checks whether the expectations as defined in the business case are satisfied. This justifies the start and execution of the project, which must be documented and remain legitimate for the duration of the project.
2. **Learning lessons:** Lessons are continually learnt from previous experiences or mistakes made by the project manager and team members. It is important to identify these lessons in order to ensure that the same mistakes are not repeated. The benefits and possible impact mistakes will have on the business will be realised if previous mistakes are not repeated. All the experience gained and lessons learnt from previous projects and current projects being executed should be passed on to other projects being executed and new projects still to be initiated.
3. **Defining roles and responsibilities:** Roles and responsibilities are defined and stipulated as part of the *organisation* theme. The project is organised by predetermining and agreeing on defined project roles and responsibilities in such a way that the executive managers, project manager, team members, users and suppliers receive the recognition they deserve.
4. **Managing stage by stage:** A project that is managed using PRINCE2 PMM will be planned, assessed, managed and controlled using a stage-by-stage approach. This ensures that the project is structured and well controlled. The shorter the stages are, the more control is ensured as the project progresses. At the end of a stage, the project plan and business case must be reviewed and the status of the project must be assessed to ensure that project stays practical and attainable.
5. **Managing by exception:** It is important that the tolerance for all deliverables be managed correctly to ensure that the project is well controlled, which is the reason that there must be certain control mechanisms and techniques put in place to ensure successful project delivery. PRINCE2 is characterised by certain tolerances for time, cost, quality, scope, risk, and benefit objectives associated with project deliverables. When these tolerances are

exceeded, they are communicated to the higher level of management.

6. **Product-based focus:** Instead of focusing on activities, PRINCE2's main focus is on defining and delivering products within the defined scope, time, cost and quality constraints in order to ensure fulfilment of stakeholder requirements and expectations.
7. **Tailoring:** PRINCE2 has the flexibility and adaptability to manage variances in relation to the importance, complexity, size and environment of a project; the competencies of the team and stakeholders; risks; issues; problems; and changes required into consideration. "The value of PRINCE2 is that it is a universal project management method that can be applied regardless of project type, organisation, geography or culture" (OGC, 2009:14). The PMM is tailored to ensure environmental applicability and execution control, based on project difficulty, importance, scale, and risk.

3.4.1.3 PRINCE2 themes

Another difference between PRINCE2 2005 and PRINCE2 2009 is that progress was added to the themes of the 2009 edition, while the component *configuration management* (as in edition 2005) was incorporated into the other themes of PRINCE2 2009. These PRINCE2 themes carry a specific importance and meaning as project management tools and approaches. These themes must be addressed continually during the course of the project. The strength of PRINCE2 lies in the manner in which these themes are linked to one another. PRINCE2 project managers use the following themes to execute the PRINCE2 processes, and to manage and organise the project effectively and professionally (OGC, 2009:17–110; Portman, 2009:4–5; Bentley, 2005):

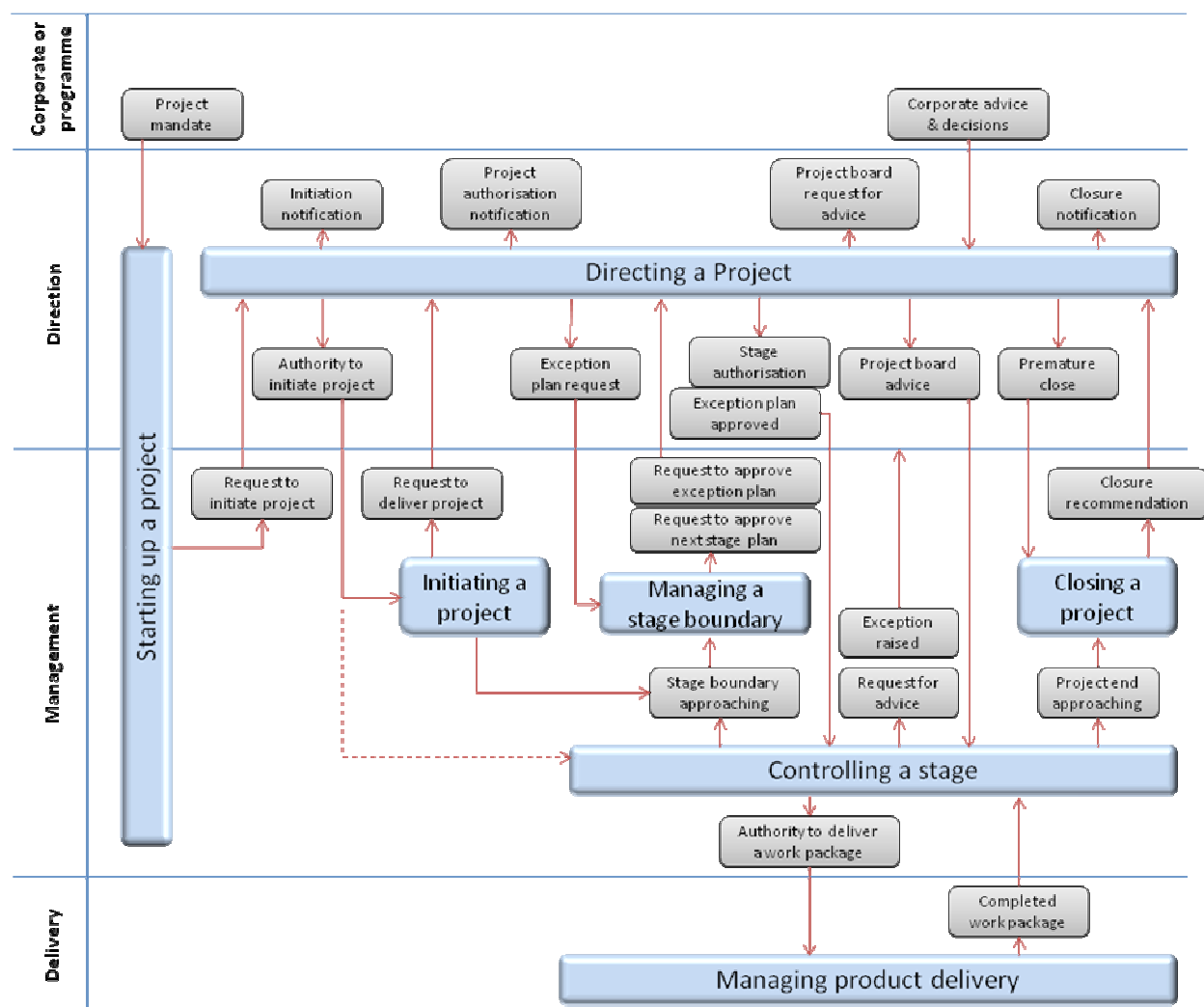
1. **Business case:** A project originally starts with an initiative or idea that might provide value to the business. The business case proposes the manner in which this initiative can offer a viable investment, and it explains what mechanisms are to be put in place in order to evaluate whether the selected initiative is achievable, viable and desirable and thereby assist stakeholders in making the right decision(s). The business case contains the reasons for undertaking the project, business operations, benefits (and value thereof), disadvantages, time and cost constraints, and the main identified risks that might affect the project. It is used as the main reference document that justifies the execution of the project in order to contribute towards the success of an organisation.
2. **Organisation:** The project's organisational structure is determined by assigning responsibilities and accountabilities. The project is organised by setting up the project executive project committee or project board, project team and other stakeholders. Responsibilities, authority and tasks, including the time-frame in which these are to be completed are delegated to the parties responsible for execution in order to manage and deliver the project successfully.

3. **Quality:** Quality demonstrates that the required expectations or specifications for the characteristics of a defined product, system, process, or service have been met or adhered too. Quality assurance ensures that products of quality are delivered successfully by meeting the expectations of the business and by subsequently enabling the business to achieve the decided benefits of the project. The quality recommendations, constraints and standards, and the responsible party are defined before the project is executed.
4. **Plans:** Planning is performed by developing and maintaining plans. Planning defines the products and deliverables that must be executed within certain time, cost and quality constraints by the person responsible. The value and means of developing and delivering products is defined to facilitate the control and communication process. This theme complements quality and explains the steps required to develop plans, which are then matched to the expectations of stakeholders across the business. Planning is done in the form of a project plan and detailed stage plan, which includes a project schedule (activities and dependencies) and WBS.
5. **Risk:** A risk is a potential problem/situation that can have a positive (opportunity) or negative (threat) impact on a project's accomplishment of objectives. The purpose of this theme is the identification, assessment and control of risks to decrease the amount of risks that could cause a project to fail. In order to manage risk, the necessary mitigation actions and preventative actions must be put in place to ensure that the risk is resolved before it has a negative impact on the project. It is also important to have contingency actions in place that prescribe what will be done when a risk arises and becomes an issue. The process includes the identification, qualification, prioritisation, mitigation, assessment and control of risks. Putting the appropriate countermeasures in place to combat risks and the impact these might have is very important.
6. **Changes:** This theme relates to change and issue management. The purpose of this theme is to identify, assess, manage and control changes made to the project. Change management ensures that changes made to the project time, budget, quality, benefit or scope are managed effectively without having unnecessary effects on the project, which might cause the failure of the project. The way in which changes are dealt with is a determining factor of a project's success. During the project life cycle, change is inevitable and every project requires issue- and change-control procedures to ensure successful delivery. According to Portman (2009:5), "PRINCE2 distinguishes between changes in the specification of a product (request for change – RFC) and changes resulting from indicated errors (off specification)".
7. **Progress:** "Progress is the measure of the achievement of the objectives of a plan" (OGC, 2009:101). The purpose of this theme is to provide a forecast for the project's continued viability and objectives; to create mechanisms to compare and monitor actual execution

tasks, objectives and achievements against the original planned time, cost, quality and scope objectives; and to control any unexpected deviations for the original plans. It furthermore ensures that products are delivered on time and within budget by tracking deliverables and products during the execution of the project. Progress assessments make it possible for the team to make the correct decisions and to execute the correct assigned tasks timeously without affecting the successful completion of the project.

3.4.2 PRINCE2 process model

PRINCE2 is driven by a process that provides organisations with a flexible, scalable and tailored PMM for different types of projects. In the process model, every process describes what must be done and what will be achieved. Bentley (2005:13) emphasises that the successful use of the PRINCE2 process model is tailoring it to the unique requirements of the project. The process model is shown in Figure 3.5.



Source: OGC (2009:115)

Figure 3.5: The PRINCE2 process model

On the left in the figure above, the PRINCE2 process model divides the processes into three different sections: *directing*, *managing* and *delivering*. Before a project is initiated the *starting up a project (SU)* process are commenced at the *directing* and *managing* sections or levels. The *managing a stage boundary (SB)* process is done for the first time at the end of the *initiation* phase and continues to be completed for each subsequent stage thereafter. The *closing a project (CP)* process is the final process to be completed during the final stage. The seven main processes of PRINCE2 and their corresponding objectives and key activities will be described in the sections that follow (Portman, 2009:3–4; OGC, 2009:113–212).

3.4.2.1 Starting up a project

The main purpose of this process is to ensure that viable project is initiated and that the prerequisites before project initiation are in place (OGC, 2009:121). The objectives of the SU process are to ensure that (OGC, 2009:121):

- the initiation of the project is justified in a business case;
- the required authorisation is given for project initiation;
- adequate information is available through a project brief that defines the project's scope;
- the different ways in which the project can be developed are evaluated and the approach to executing the project is selected;
- people are appointed as part of the project team or as project managers who will be responsible for executing the work required during project initiation;
- the tasks to be completed during project initiation are planned; and
- time is not spent on initiating a project that is based on unstable assumptions and expectations regarding constraints, scope, time, and acceptance criteria.

Starting up a project is the process to be executed before the project is initiated. During this process, the project team is appointed and the interim business case or project brief is developed that justifies the execution of the project. The project brief explains the reason and requirements of the project, the benefits, the risks and issues, the cost and the time-frame. The effort, time and cost involved during the SU process will vary significantly from project to project, as every project has its own unique characteristics. The next stage of the project is planned and the approach to be taken in order to deliver the products is decided upon. In some cases, alternative approaches to delivering the project are decided upon once the approach has been finalised. The project board authorises the start of the initiation stage. The key activities to be included during this process are (OGC, 2009:122):

- *Appointing an executive and project manager:* The executive and project manager is first appointed to ensure that the project is justified and that project will be managed on a day-to-day basis (OGC, 2009:123).

- *Capturing previous lessons learnt:* The previous lessons learnt are captured to ensure that the same mistakes made during previous projects are not repeated. This includes the “weaknesses or strengths of the processes, procedures, techniques and tools used, when they were used, how they were used, and by whom” (OGC, 2009:124).
- *Designing and appointing a project management team:* During this process, every person involved in the project is assigned certain responsibilities and accountabilities when the reporting and communication structure is put in place. The right people with the right skills need to be assigned with the knowledge to make informed decisions that will direct and manage the project to successful completion.
- *Selecting the project approach and assembling the project brief:* A decision must be made regarding the manner in which the project will be approached before planning can commence. The manner in which the work will be completed depends on client practices, guidelines, standards or expectations, which must be documented in the project brief.
- *Preparing the outline business case:* The main purpose of the business case is to define the reasons for executing the project by explaining the value it can bring to the organisation. The outline business case is only a high-level framework at this point and will be developed in detail during the *initiating a project* (IP) process.
- *Planning the initiation stage:* The initiation stage of a project requires many resources and is normally time consuming. The initiation stage is approved by the project board and project team to ensure that it is structured and directed towards project success.

3.4.2.2 Directing a project

The main purpose of this process is to enable the project board to practise project controls and to make important decisions while delegating the daily project management and execution to the project manager (OGC, 2009:135). This makes the project board responsible for the overall success of the project. The objectives of the DP process are to ensure that (OGC, 2009:135):

- authority has been provided to initiate the project;
- authority has been provided to deliver the product(s);
- the project remains viable, and that direction and control are provided throughout the duration of the project;
- programme and corporate management has an interface to the project;
- authority has been provided to close the project; and
- plans are managed and reviewed to realise the post-project benefits.

This process is executed by the project board, which runs from the *SU* process to the finalisation of the *CP* process. This process prescribes the manner in which the project board must control the project and authorise a stage plan. Furthermore, the project board has the

ability to provide *ad hoc* guidance on the way in which a project should be closed down. The aim of the process is the successful execution and delivery of the products, services or deliverables of the projects. The key activities to be executed during this process are (OGC, 2009:136):

- *Authorising initiation:* The project board needs to make a decision on whether the project can be initiated by ensuring that the investment the project will bring to the organisation will be of value. It is expensive for a project to be initiated, which is the reason that all activities for the initiation stage must be planned, controlled and monitored (OGC, 2009:136).
- *Authorising the project:* The aim is to determine whether the organisation should proceed with the project. This activity must be executed in parallel with *authorising an exception plan or stage* activity and is prompted by the project manager, who requests authorisation to execute and deliver the project. A project is prematurely closed out when it has not been approved by the project board (OGC, 2009:137).
- *Authorising an exception plan or stage:* Other than in the last management stage, the project board authorises the execution of the next stage's plan after reviewing the progress and performance of the current stage for all management stages. Exceptions to stage plans or the project plan that occurred during the execution of a stage must be presented to the project board for approval using an exception plan in order for the approved exception plan to become the new base-line stage or project plan. Exceptions in work packages are managed in the controlling a stage (CS) process by the project manager. In some cases, deviations from the original project plan must be approved by programme or corporate management (OGC, 2009:139).
- *Giving ad hoc ad hoc direction:* The project board can provide collective or individual member guidance and advice to the project team at any given time. The need for project manager direction and guidance will most like occur as the project reaches stage boundaries and during project initiation (OGC, 2009:141).
- *Authorising project closure:* To ensure that promises made to the client or organisation in the initiation documentation and project plan are fulfilled, the project must be formally closed off in a controlled manner. This will show what has been achieved by the project and the deviations of the project from the initial project plan and scope (OGC, 2009:143–144).

In all of the activities above, the project board can assign project or organisational assurance in order to execute some of the assessing and reviewing functionality. The project board manages the project by considering exceptions as presented by the project manager, and acts as a communication channel by communicating all relevant project and progress information to corporate management.

3.4.2.3 Initiating a project

The main purpose of the IP process is to create a firm basis for the project that will inform the business of what must be done to deliver the products successfully before they commit to spending the budget (OGC, 2009:149). The objectives of the IP process are to ensure that there is a general understanding of (OGC, 2009:149–150):

- the benefits expected, the reason for executing the objectives, and the related risks;
- the products to be delivered as part of the scope of the project;
- the time-frame and the manner in which the products will be developed, and the associated costs to deliver them;
- the people involved in making the necessary decisions;
- the manner in which the required quality of products will be achieved;
- the manner in which the base-line of the project will be established and controlled;
- the manner in which the project's progress will be controlled and monitored;
- the persons who require certain information, as well as the date and the format of such information; and
- the tailoring of the PMM to suit the project environment.

After the SU process has been completed, the project brief is finalised into a business case on which all affected parties agree. This process ensures that all parties involved are aware of what products and services will be delivered, by whom they are to be delivered and the cost, time and quality constraints associated with each of the products. The roles, responsibilities and authority of each person involved in the project are determined during this process. Project controls and a project file are also developed before the project board authorises the next phase of the project. A plan to be executed for the next phase (*directing a project* – DP) is also completed before authorisation is provided. The key activities to be executed during this process are (OGC, 2009:150):

- *Setting up the risk management strategy*: This strategy explains the risk tolerance levels, risk management roles and responsibilities, procedure on dealing with risks, timing of risk activities, reporting (register) requirements, tools and techniques, and the objective of applying a certain risk management approach (OGC, 2009:150).
- *Setting up the configuration management strategy*: This strategy will define the manner in which control is maintained over all products of the project, and it will define the format and composition of the configuration item records that need to be maintained. Optimal control is attained when a product is broken down into separate components that that can be modified, installed or replaced individually. The relationship intricacy between the products and the project's importance affect the level of control put in place.
- *Setting up the quality management strategy*: The main purpose of this strategy is to ensure

that the quality expectations and standards are documented and agreed upon before the project is executed, including the manner in which quality will be measured, assessed and maintained as the project progresses and when the project reaches completion (OGC, 2009:153).

- *Setting up the communications management strategy:* In cases in which a stakeholder engagement procedure is required, which describes communication strategies, evaluation methods of communication success, desired relationship, and the types of stakeholders, it must be documented as part of this strategy. This strategy defines the manner in which the project team sends and receives internal and external communications from stakeholders and the rest of the project environment, including the type of information and format in which it must be provided to senior management if it exists (OGC, 2009:155–156).
- *Setting up project controls:* “Effective project controls are a prerequisite for managing by exception” (OGC, 2009:157). The control levels and mechanisms required by the project manager and the project board need to be established and agreed upon in order to enable the effective management of the project (OGC, 2009:156–157). This will include controls on managing risks, scope, quality and other project complexities.
- *Developing the project plan:* Planning is a group effort that needs to occur in a brainstorming or workshop session(s) with the project manager, project team, suppliers, users and other stakeholders. In this activity, the time scale, milestones, resource requirements and dependencies are captured, before large project expenditure is undertaken, which will be used by the project manager and project board to manage and control the project’s progress (OGC, 2009:159).
- *Refining the business case:* After the project plan has been developed, the initial business case developed during the SU process is revised to reflect the calculated cost and estimated time scale, which will be used by the project board to authorise the execution of the project. Any newly identified or updated risks are also revised in the risk register (OGC, 2009:161).
- *Compiling project initiation documentation:* This activity entails the collection of all initiation documentation used to obtain authorisation for the execution of the project. This documentation is made available for guidance and information to all stakeholders involved in the project as it contains the why, who, how, what, when, where and how much answers to possible questions arising as the project progresses. This collection of documents is used to monitor, compare and approve the actual project progress and performance against what was predicted.

The IP process allows the project board as part of the DP process to authorise the continuation of the project by checking whether the project aligns with the strategic and business objectives

of the organisation. As soon as the project has been initiated, it must be controlled from start to finish.

3.4.2.4 Controlling a stage

The main purpose of the CS process is to allocate tasks to be completed by the project team, monitor the work completed, manage risks and issues, undertake exception and progress reporting to the project board, and take the necessary corrective actions to drive the project to successful completion (OGC, 2009:167). The objectives of the CS process are to ensure that (OGC, 2009:167–168):

- The focus is on the products to be delivered. Any deviation from the original products agreed upon at the beginning of the stage is monitored to avoid divergence from the focus that might cause scope creep.
- The issues and risks of the project are controlled.
- The business case is reviewed on a regular basis.
- The defined and accepted products are delivered according to quality, time and cost constraints to achieve the defined business benefits as stipulated in the business case.
- The team delivers within the acceptable constraints and tolerances.

The stages of the PRINCE2 project must be broken down into sub-processes that explain the manner in which each stage will be controlled and executed. It includes the manner in which risks (potential problems), issues (current problems), changes and other unforeseen circumstances will be dealt with to ensure that these do not have a negative impact on the project. In order to manage control effectively, it is important to report on the progress of a project in order to identify possible and current problems and to identify resolution actions before they become a threat to the successful delivery of a project. The process of escalating identified risks, issues and changes to the project board must also be formalised during this process. The activities to be completed during this process are (OGC, 2009:168):

- *Authorising work packages:* A work package should contain a grouping of activities from the initiation documentation, project plan, and stage plans to produce one or more products. When more than one work package is required to create a certain product, the product must be broken into sub-products that include the sub-product description and client expectations. Each work package must be authorised by the project manager, as disorganisation would otherwise result (OGC, 2009:168).
- *Reviewing work-package status:* The status of the work package must be regularly assessed and measured to ensure that the project is on track. The regularity of the status reviews will depend on the reporting frequency and expectations of the stakeholders (OGC, 2009:170).

- *Receiving completed work packages:* Confirmation must be provided by the team members responsible for completing assigned work in order for it to be approved. If there were any changes to the allocated work or product, it must undergo the change control process (OGC, 2009:172).
- *Reviewing stage status:* The purpose of this activity is to maintain and monitor project progress accurately on the work being completed in order to ensure control by checking what happened, what was expected to happen, and the possible effect of the current state on the next stage as the project progresses. Reviewing the stage's status can occur when it is initiated by the project board, during issue and risk analysis, or as stipulated in the stage plan (OGC, 2009:173).
- *Reporting highlights:* Summarised stage information and project status must be reported to the project board in the format and as frequently as stipulated in the communication management strategy (OGC, 2009:175).
- *Capturing and examining project issues and risks:* Risks and issues can be raised by any stakeholder on corporate, programme, or project management level and they must be captured in a reliable manner, as risk can be identified and issues may occur as the project progresses (OGC, 2009:176).
- *Escalating project issues and risks:* The project manager must escalate all issues and risks to the project board that threaten the tolerance levels as stipulated by the project board, after applying all possible corrective actions within his/her control. After escalating the risk or issue, an exception report is developed by the project manager that provides supporting information on the escalated risks and issues, which will also include recommended corrective actions. The sooner risk and issues are escalated, the more time can be spent on developing and executing corrective actions (OGC, 2009:179).
- *Taking corrective actions:* This activity is done when the status of the stage is reviewed by the project team and project board. The main purpose is to implement corrective actions that are within stage and project tolerance limits that will not cause any project plan deviations.

The SU process is normally executed for the first time after the project board has authorised the project. This process will be used for each delivery stage of the project by the project manager to manage the stage on a daily basis. At the end of every stage, except for the last stage, the *managing product delivery* (MP) process will commence.

3.4.2.5 Managing product delivery

The CS process views the project and products from the perspective of the project manager, while the MP process is viewed from the team manager's perspective (OGC, 2009:185). The

main purpose of the MP process is to control the relationship between the project manager and the team manager(s) by placing defined acceptance, execution and delivery requirements on objectives and activities that must be completed. The objectives of the MP process are to ensure that (OGC, 2009:185):

- product-related tasks allocated to team members are approved and agreed upon;
- the team members, managers and suppliers agree upon and understand what must be produced within the cost and time constraints and what effort will be involved in completing the work;
- the planned objectives and product are completed within the acceptable tolerance levels and expectations; and
- expectations are managed by providing accurate progress information timeously and according to the agreed frequency.

This process describes the tasks that must be completed by the parties responsible and the manner in which these will be managed to ensure that the product is delivered within cost, time and quality constraints as specified in the project plan and business case. The parties responsible (team members and third parties such as suppliers and contractors) will execute these activities independently from the project manager. The activities to be completed during this process are (OGC, 2009:186):

- *Accepting a work package:* There must be a clear understanding between the project manager and the project team as to what must be delivered within certain constraints and whether it is achievable with the resources that are available. Reporting requirements, the manner in which to provide feedback on work completed, the deadlines for feedback on work completed, and any further procedures to be followed must be stipulated and understood (OGC, 2009:187).
- *Executing a work package:* The defined expectations and requirements of the accepted work package are executed and monitored until completion. During execution, the project team must stay within the accepted tolerance limits. Should tolerance limits be threatened the team manager should report the situation to the project manager. The team manager may execute corrective actions provided that the tolerance limits set by the project manager are not threatened (OGC, 2009:188).
- *Delivering a work package:* The project manager must be informed as soon the requirements of the work package have been satisfied (OGC, 2009:190).

3.4.2.6 Managing a stage boundary

The main purpose of the SB process is to provide the project board with sufficient information in order to review the status of the current stage, make informed decisions, accept and approve

the next stage plan, confirm acceptability of risks and continued business justification, and re-evaluate the project plan (OGC, 2009:193). The objectives of the SB process are to (OGC, 2009:194):

- provide the assurance to the project board that the products associated with the current stage plan have been completed and accepted;
- ensure that the stage plan for the next stage is prepared;
- review and update the original initiation documentation, such as the project plan, business case, project team structures and responsibilities, and project approach strategies;
- assist the project board by providing them with sufficient information to conduct continuous viability assessments of the products to be delivered – this also includes the defined risk exposure and tolerance;
- record lessons and other relevant project information to assist with the execution of other stages or projects;
- request authorisation in order to begin the next stage; and
- prepare an execution plan as stipulated by the project board, and obtain acceptance for replacing the current stage's plan or project plan with the exception plan in a situation in which exceptions are required during the SB process.

The project is managed in local units or stages. Each stage has its own series of activities that must be executed by the parties responsible in order to close-out a specific stage and thus for the next stage to commence. The process explains the manner in which a stage must be closed down in order to complete the stage. The next stage of the project is planned before the close-out of a preceding stage in order to ensure that the project retains its momentum and that the parties responsible are correctly assigned to the activities of the next stage for execution. Before the team moves on to the next stage, the progress must be updated regarding the business case, project plan, risk register, change register, issue register and other relevant documentation. The process also explains what should be done if the stage exceeds its tolerance levels and the manner in which the completion of a stage should be reported on. From the perspective of a project manager, the process explains what activities must be executed to close the stage and what activities must be considered to start preparation for the next stage. The activities that will be completed during this stage include (OGC, 2009:194):

- *Planning the next stage:* At the end of the current stage, the stage plan for the next management stage is planned by all the stakeholders (project manager, project board, project assurance and members of the team) involved. During the final management stage, the necessary project closure activities must be included (OGC, 2009:194).
- *Updating the project plan:* After the completion of a stage, the project plan is updated to reflect the progress made. The forecast duration and cost of the stage plan or exception

plan for the next stage is included in the update to ensure that the project board is given a comprehensive view of the progress made in the project (OGC, 2009:196).

- *Updating the business case:* The business case must be updated, reviewed and amended regarding the changes that occurred in the project environment. It is the responsibility of the project manager and board to ensure that the project has continuous business justification and that it remains viable by ensuring that the benefits stipulated in the business case are achieved. When updating the business case, the project manager must consult the executive(s) before presenting it to project board for approval, as the executive(s) owns the business case (OGC, 2009:197).
- *Reporting stage completion:* When the stage reaches completion, the project manager must provide feedback to the project board on the project's progress and its ability to meet the expectations stipulated in the business case and project plan (OGC, 2009:199).
- *Producing an exception plan:* In the CS process, the *escalating project issues and risks* activity produces an exception report, which will result in the project board requesting an exception plan that will respond to the activities that caused the project to deviate from the agreed tolerance levels in order to get the project back on track (OGC, 2009:201).

The SB process should be executed close to the end of each stage, as it provides the way by which a defined exception process can be approved and implemented by the project team. The process is however not executed towards the end of the final stage, as the tasks to review the performance of the final stage are included as part of the CP process, which reviews the whole project.

3.4.2.7 Closing a project

The main purpose of the CP process is to ensure that project's products and objectives as documented in the initiation documentation have been achieved, delivered, confirmed and accepted (OGC, 2009:205). The main objectives of the CP process are to (OGC, 2009:205):

- ensure that user acceptance of the project's products have been verified;
- ensure that the host site has the ability to support the products when the project is closed out;
- review the project's performance against the base-lines of the project;
- identify and assess benefits that have already been determined, update and plan a review of the unrealised benefits; and
- ensure that all remaining issues and risks are addressed through recommendations regarding preventative and corrective actions.

During a process, a project must be formally closed out and accepted by the project board. The business case must be assessed to ensure that what was promised to be completed has been done. If certain aspects or tasks have not been delivered according to the quality standards (tolerance levels) stipulated, the necessary proof must be provided of the manner in which the tasks/activities were managed and accepted during the execution of the project. After the activities have been completed and the project has been closed out, reviewed, evaluated and accepted by the project board, the project manager is relieved of his/her commitments. The activities to be executed during the closing of a project are (OGC, 2009:206):

- *Preparing planned closure:* After the project manager has determined that all expectations and requirements have been met within the project constraints as stipulated in the project plan and business case, the project manager can suggest that the project be closed out (OGC, 2009:206).
- *Preparing premature closure:* In situations in which the project is closed out prematurely by the project board, it is the project manager's responsibility to save all the valuable work that has been completed and to discuss all the risks, issues, problems and gaps that may result with senior management (OGC, 2009:207).
- *Handing over products:* Before the project closes, the completed product(s) must be handed over to the client for operational use. The products can be handed over in a phased approach with a number of releases or in one release, where the project as a whole is handed over to the client. After hand-over, the benefit review plan must be updated to identify whether there are any lessons learnt that will be useful for other projects. In cases in which the project has been closed prematurely, the project board must advise regarding whether products that have been completed must be transferred to the client (OGC, 2009:208).
- *Evaluating the project:* The main purpose of this activity is to determine the extent of success of the project through comparison of what was promised in the business plan and project plan and what has been delivered.
- *Recommending project closure:* Project closure can be recommended to the project board as soon as the project manager has confirmed that closure can commence after all hand-overs, sign-offs and evaluations have been completed (OGC, 2009:211).

PRINCE2 has a clear end to a project. The activities to be completed during the CP process must be part of the stage plan for the final stage to be executed by the project. There may be some actions to be completed after sign-off has been received, which must be documented and follow-up action recommendations must be made to address these actions (OGC, 2009:206).

Different from PRINCE2 2005, the planning process has not been included separately in

PRINCE2 2009. Instead, the process has been incorporated into each of the seven PRINCE2 management processes. After explaining PRINCE2 2009 in the previous sections, this PMM will be evaluated in the section that follows.

3.4.3 Evaluation of PRINCE2

PRINCE2 was originally designed by the OGC as a government standard for developing and delivering IT projects. This makes it more applicable to IT environments than other traditional PMMs, although PRINCE2 was found to be successful in other environments as well.

Application area

PRINCE2 has spread and been adopted as one of the most acceptable project management approaches by private and public organisations internationally (Huijbers *et al.*, 2004:13) and it has been translated into many different languages (Clarkson, 2010:3). Some of these international organisations include Barclays, Standard Bank, Rolls Royce, Norwich Union, British Medical Association, ALTRAN, the UK Police Force, and DHL.

Clarkson (2010:5–7) explains the business benefits that PRINCE2 offers for commercial organisations, public-sector organisations, voluntary sector organisations and the individual. Commercial organisations using PRINCE2 are at an advantage to the public sector during tender processes, as PRINCE2 is generic and a well-known general language is used to communicate across all industry sectors effectively. Public-sector (commissioning) organisations working with supplier organisations with effective PRINCE2 utilisation will benefit from this, as it will result in successful and better project delivery than the usual supplier organisation that does not use a specific PMM. In voluntary-sector organisations such as charity and humanitarian agencies, PRINCE2 will contribute to successful project delivery that will result in the most direct measurable improvement and that is the improvement of the organisation as a whole. At individual level, a PRINCE2 certification can increase the competency of a project manager to deliver a project successfully, as PRINCE2 experience along with a PRINCE2 qualification will increase PRINCE2 competency.

Advantages

PRINCE2 has certain advantages and strengths because it (Webber & Webber, 2009:18–21; Jones, 2007:224; Van Bon & Verheijen, 2006:215; OGC, 2010:48; Charvat, 2003:79–80):

- has forms and templates that provide uniformity of projects;
- is an open standard that can be utilised by anyone without paying any fees (this excludes purchasing the book);

- promotes effective usage of management's time, as management is only involved when necessary because issues are raised according to the principle of "manage by exception";
- has a more detailed project execution approach than that of PMBOK, as planning, allocation of tasks, result validation, and progress reporting is done at a deeper level;
- is scalable – meaning that PRINCE2 is applicable to any type of project with different levels of complexity, budget, scope, etc.;
- has a detailed and formal set of client requirements, expectations and constraints, which include measurements for delivering quality work;
- acts as a guide to assist project managers and other stakeholders through a sequence of processes and steps to implement a project from start to finish;
- has seven processes that are more detailed than the five project phases of PMBOK;
- consists of a standardised manual that simplifies training and new team member integration and acceptance;
- provides controlled change management in terms of its return on investment and investment as a whole;
- provides constant user involvement to ensure that the delivered product satisfies environmental and functional requirements;
- provides effective control over the resources of the project;
- provides constant risk and quality management throughout the project's lifespan;
- provides constant control over changes to products and to the products as a whole;
- provides a controlled start, finish and progress of a project;
- reduces the risk of a project by providing a proven process framework, clear governance, and advice regarding best practices that were found to be valuable;
- is easily integrated into other industry-specific life cycles;
- promotes an economical reporting approach and communication by providing common vocabulary that can be read and understood by all members involved;
- ensures that stakeholders are properly represented during the planning and decision-making processes;
- provides a structure of accountability, authority and delegation to ensure that each individual involved understands his/her role and responsibilities, as well as the roles and responsibilities of other team members;
- is focused on delivering products and clarifies what must be delivered by the project as a whole;
- improves communication and control by the plans that are developed to satisfy different management level requirements and expectations;
- ensures that the project is viable throughout the duration of the project;
- promotes continual improvement and learning processes;

- facilitates the auditing and quality assurance of work done during the project;
- is supported by experts and internationally accredited training; and
- focuses on project results.

PRINCE2 is a sequential PMM that follows a waterfall process model, which is slow to react to change (OGC, 2010:x) but has the advantage that it integrates the organisation, different plans and useful controls. When applying PRINCE2 as a PMM, the project manager and team can choose which test method can be applied as part of the project quality plan (Huijbers *et al.*, 2004:14). According to Clarkson (2010:9), PRINCE2 is a risk management tool for organisations that require efficient transformation or change. PRINCE2 is recognised in the industry as a generic PMM with a universal language that is used across geographical borders. It is teachable and repeatable, tried and tested across different industries (Clarkson, 2010:9). Another strength of PRINCE2 is that the next stage is only planned as soon as the current stage has been completed (Graham, 2010:86).

PRINCE2 embraces the importance of a business case as the main document that is updated and referenced continuously to ensure that what was promised to the client is delivered. The PRINCE2 project environment (see Figure 3.5) and *tailoring* (PRINCE2 principle) emphasise that PRINCE2 can be tailored to suit different project types and different project sizes with different complexities in different environments.

The principle of *managing stage by stage* with the CS and SB processes functions in harmony to resolve risks, issues, and changes quickly and to report on the project's progress effectively to all stakeholders involved. In the managing stage-by-stage principle, according to which the next stage is planned before closing the current stage, PRINCE2 highlights incremental development, which encourages combination with ASDMs, since one of the major focuses of ASDMs is incremental development.

One other aspect that makes PRINCE2 different from other PMMs is that it requires an executive project committee to be established that must govern the project under the theme *organisation*. In contrast with PMBOK, PRINCE2 does address issue management under the theme *change*. The theme *change* also ensures that change is managed and controlled correctly throughout the lifespan of the project.

Many South African organisations struggle to provide direction from senior management, but PRINCE2 ensures that direction is provided by senior management if the DP process is incorporated into the business. Another positive attribute of PRINCE2 is the strategies (quality,

risk, etc.) developed as part of the IP process, which ensures that direction is provided early in the project and dictates the manner in which the project will be managed.

PRINCE2 has a formalised project close-out process, which includes an evaluation process to assess whether the project was completed successfully. This is determined by whether the promises made to clients in the business case were satisfied.

Disadvantages

PRINCE2 implies that with continual planning, it can account for all uncertainties. This is not the case – no PMM can ensure that everything will happen as predicted.

PRINCE2 has certain weaknesses that must be considered (Van Bon & Verheijen, 2006:215; Jones, 2007:22; Charvat, 2003:80):

- PRINCE2 is assigned by a manager as the PMM of choice without him/her knowing the reasons for this or without being familiar with PRINCE2. Managers then use this in some cases to shift blame when projects start to fail.
- Because PRINCE2 is scalable, many organisations drop so many aspects of PRINCE2 that they might not be following PRINCE2 at all anymore. (This is referred to as PINO – PRINCE In Name Only.)
- Although PRINCE2 is scalable, it still has “too much structure overlaying the project” in comparison with ASDMs.
- It is not a complete answer to managing all projects – people must continue to enhance the PMM.
- PRINCE2 may not be tailored correctly, as it may be interpreted and applied too rigidly.
- Human and individual behaviour management is essential to successful delivery of projects.
- The business case lacks insufficient detail regarding investment appraisal.

There is an assumption that when PRINCE2 is applied in practice “that every part of every process must be checked off” and that learning to implement PRINCE2 is difficult (Fletcher, 2006:139). The greatest disadvantage of PRINCE2 is that too much documentation is required when it is applied in practice (OGC, 2002:27). This might cause a communication breakdown and blame shifting when something in the project goes wrong. Since PRINCE2 requires a large amount of documentation to be completed, changes to documentation will cause a great deal of time to be wasted and money unnecessarily spent.

Adding *communication* as an eighth principle would make the foundation of the PMM much stronger. A PRINCE2 principle guides project managers, the team and executive managers in

delivering IT projects successfully. In order to be successful, a project requires integrated and transparent communication embedded within the foundation of the PMM.

PRINCE2 has a certain language and expressions associated with it, and if it is not understood by the majority of the organisation, PRINCE2 will most likely fail. All the stakeholders involved need to understand the PRINCE2 language (Huijbers *et al.*, 2004:14), which in most circumstances it is not the case, as training is expensive and not all stakeholders have the time to study PRINCE2. Like PMBOK, PRINCE2 is complex and a project manager must undergo intensive training to utilise the PMM correctly. He/she also runs the risk of bearing the blame for a failed project if PRINCE2 was applied incorrectly, although external factors might have caused the project failure. In the midst of organisational chaos, PRINCE2 will suffer because it will be impossible to adhere to the complex processes PRINCE2 provides. According to Moore (2009:149), PRINCE2 can “dilute the dimension of human engagement”. Furthermore, it must not be used to “manage any initiatives critical to the strategic success of an organisation”.

It is still not clear whether PRINCE2 processes are executable in a constantly changing environment because each environment changes differently. Incorrect prediction of which processes must be added or removed from PRINCE2 to tailor it to its environment can lead to project failure, as the immediate project environment of today can be entirely different from one a month ago when the PMM was tailored. Furthermore, PRINCE2 might be only partially accepted by organisations. For example, if an organisation does not accept the DP process, this will result in senior management not governing the project or giving direction, which could ultimately result in an unsuccessful project. It would thus be wise to conduct an impact analysis before PRINCE2 is partially accepted into an organisation. The way in which PRINCE2 is applied can lead to project failure if it does not suit the project and business environment. From this, it appears that PRINCE2 may not be as flexible and suited to tailoring as its developers claim. This claim must still be verified by practitioners.

Integration with other methodologies

The statement by Huijbers *et al.* (2004:14) that “PRINCE2 does not need to be coupled to other methodologies or project techniques to complete itself” is untrue, as every PMM has flaws and areas that need improvement. PRINCE2 can be integrated with other methodologies to make it more complete than what it is currently.

The OGC suggests that projects can be managed and delivered using a combination of PRINCE2 and DSDM (OGC, 2010:198; Müller, 2009:26). PRINCE2 can be combined with ASDMs to create an effective solution (Read & Properjohn 2008:1), although Scrum and LD

might pose some challenges, as Scrum covers more management issues, while LD's underlying principles are quite different from that of PRINCE2 (Kelly, 2008:37). That PRINCE2 has a whole chapter (OGC, 2009:215–231) dedicated to tailoring it to adjust and fit to its business and project environment makes it more than applicable for use in conjunction with ASDMs because like PRINCE2 the main focus of ASDMs is to change and adapt to their environment. However, according to Kelly (2008:37), the integration of ASDMs and PRINCE2 will result in both being compromised because when they are combined certain individual strengths will fall away. Integration with other methodologies emphasises that PRINCE2 and other ASDMs, such as DSDM, alone are not sufficient to deliver IT projects. For this reason, this study sought to determine whether PRINCE2 could be integrated with the best practices of other ASDMs and PMMs to deliver IT projects in a constantly changing project and business environment.

3.5 Agile Project Management

Agile Project Management is a paradigm shift from the traditional plan-then-execute-project paradigm that embraces the fundamentals of the normal four-stage (initiate, plan, execute, close-out) project life-cycle phases to a new five-phase (envision, speculate, explore, adapt, close) project life cycle, as explained by Highsmith (2010). Traditional PMMs aim to prevent change by extensively planning and documenting as much as possible before the system is developed, while APM accepts that change is inevitable and that it is not to be avoided but managed (Karlesky & Voord, 2008:2). “Agile Project Management lets software project managers and employees alike adapt to changing circumstances, rather than to improve rigid formal controls as in traditional linear development methods” (Augustine, Payne, Sencindiver & Woodcock, 2005:85).

In the ever-changing marketplace, change is inevitable. This is the reason that organisations today have a desperate need for a PMM able to adapt to its environment. Many organisations make the mistake of applying structured PMMs simply because they worked for other organisations. Practitioners need to understand that many businesses have been operating for many years with many business processes, techniques and tools in place, which have cost a great deal of money to develop and implement. Implementing a structured PMM in such an organisation will have a major impact on the business and there is general resistance to change, which must be managed. Often, the person(s) responsible for incorporating the structured PMM expects that the rest of the business processes and tools must adapt to the PMM, from which many difficulties arise, as many business processes cannot be adapted because of political and other issues, debated and decided upon over several months (Hass, 2007:1–2). For this reason, APM was included as part of this study to investigate whether it could address these challenges. Agile Project Management will mainly be described with

reference to Highsmith (2010).

Agile project management has four focal points (Highsmith, 2010:1):

- “opportunities created by the agile revolution and its impact on product development;
- values and principles that change APM;
- specific practices that embody and amplify these principles; and
- practices to help entire organisations, not just project teams, embrace agility.”

Software development teams are revolutionising where they have to adjust from an “anticipatory to adaptive styles of development” (Highsmith, 2010:5). To build innovative software products, APM has five business objectives (Highsmith, 2010:10):

- *continuous innovation*: deliver on current client and business expectations;
- *product adaptability*: deliver on future client and business expectations;
- *improved time to market*: meet market windows and improve the return on investment;
- *people and process adaptability*: react rapidly to required product and business changes; and
- *reliable results*: support business growth and profitability.

3.5.1 Agile Project Management values and principles

The three core APM values (Highsmith, 2010:14–17) summarise the Agile Software Development Manifesto (see Section 2.3.1; Beck *et al.*, 2001; Highsmith, 2010:16) and Declaration of Interdependence founded by the Agile Project Leadership Network (Highsmith, 2010:14). The Declaration of Interdependence (Highsmith, 2010:14) will be presented first, after which the three core APM values will be provided:

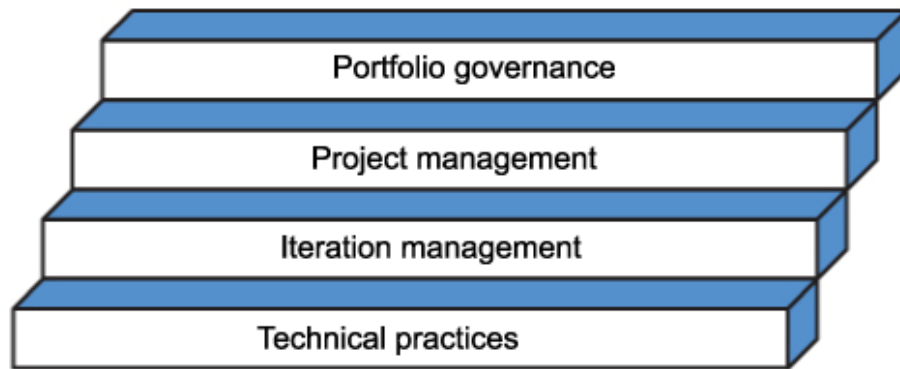
- “We increase return on investment by making continuous flow of value our focus.
- We deliver reliable results by engaging customers in frequent interactions and shared ownership.
- We expect uncertainty and manage for it through iterations, anticipation, and adaption.
- We unleash creativity and innovation by recognising that individuals are the ultimate source of value, and creating an environment where they can make a difference.
- We boost performance through group accountability for results and shared responsibility for team effectiveness.
- We improve effectiveness and reliability through situationally specific strategies, processes, and practices.”

The three core APM values (Highsmith, 2010:14–17) include the following:

1. *Delivering value above meeting constraints*: This value provides “a focus for rethinking how we measure performance on projects” (Highsmith, 2010:27). Traditional project managers focus on delivering according to the time, cost and quality requirement constraints as defined in the project scope, while agile project managers focus on delivering value and “constantly asking questions about whether different renditions of scope are worth the value they deliver” (Highsmith, 2010:27).
2. *Leading the team above managing tasks*: “Agile leaders lead teams, non-agile ones manage tasks” (Highsmith, 2010:47). The main focus of APM is to build self-organising teams and to manage them with a “lead-by-serving mentality”. There are four major themes related to constructing teams:
 - constructing self-organising project teams;
 - leadership;
 - collaborative teamwork (including decision-making); and
 - client collaboration.
3. *Adapting to change above conforming to plans*: Traditional project managers see the “plan as the goal” and focus on “following the plan with minimal changes”, while agile project managers see “client value as the goal” and focus on “adapting successfully to inevitable changes” (Highsmith, 2010:63). The project plan becomes the means to achieve certain goals not the goal itself when quality and client value are the main objectives. Although the constraints defined in project plans are very important, project plans are not sacred; “they are meant to be flexible; they are meant to be guides” that do not restrain the team (Highsmith, 2010:63). Agile Project Management principles derived from the adaptive principle statements are summarised as (Highsmith, 2010:64):
 - Accept change (uncertainty) and respond accordingly rather than follow outdated plans.
 - Adapt processes and practices as necessary.

3.5.2 Agile enterprise framework

Agile system development methodologies have to suit and adjust to different organisational structures that consist of different levels. The agile enterprise framework breaks a organisational structure down into possible management levels that would cater for the acceptance of ASDMs into the structure of an organisation. This is shown in Figure 3.6.



Source: Highsmith (2010:78)

Figure 3.6: The agile enterprise framework

In this framework, ASDMs can be positioned for each level as this assists organisations in developing hybrid ASDMs to adapt to each project, organisational environment and requirements. The structure furthermore motivates less adaptability and flexibility at the higher portfolio governance level and more at the lower technical practices level.

At the portfolio governance level, organisations evaluate whether projects address the two main concerns of executive sponsors, namely; investment and risk (Highsmith, 2010:79). Governing frameworks and mechanisms can be created to address these two major concerns in order to ensure that valuable projects are provided to the sponsors with a return on investment within acceptable uncertainty and risk tolerance levels.

The project management level provides guidance on managing all the different projects within an organisation. This level's main concern is the "overall project release activities, assisting coordination amongst multiple feature teams, and managing the project externals" (Highsmith, 2010:79). Furthermore, other practices such as contract and risk management can be incorporated into this level. It is important to realise that a large project within an organisation can have an overall project manager who manages several teams and each team has its own iteration manager.

The individual iterations within a project are managed by the iteration management level, where leadership in a team, planning, and execution is viewed as the main focus. The difference between project and iteration management is that other than managing the project as a whole, project managers manage stakeholders outside the core project team and releases, while iteration managers manage internal team members and the individual iterations.

The technical practices level assists in defining a set of technical practices to be used in projects or iterations within an organisation. The technical level is crucial to delivering projects

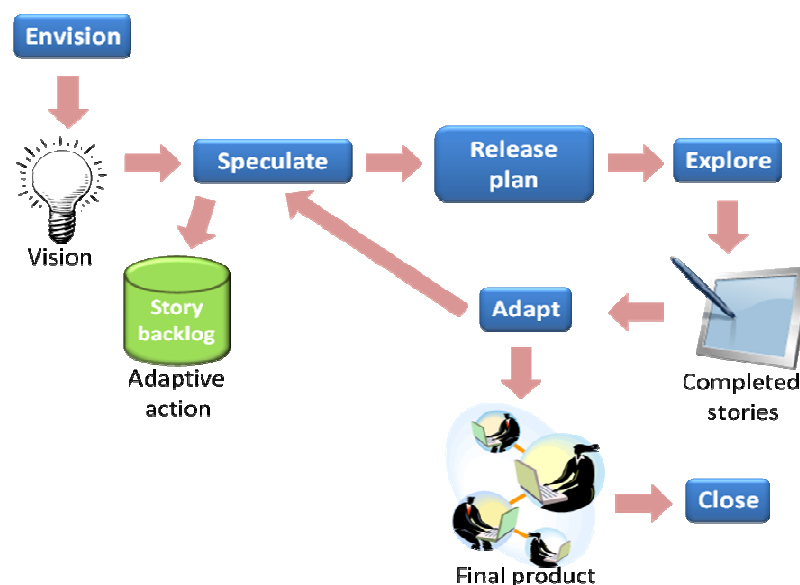
successfully, which is the reason that it is important for an organisation to transform its technical practices when ASDMs are implemented. Furthermore, the technical practices level makes APM more applicable to the variety of products and project types within an organisation.

3.5.3 Agile Project Management Model and Delivery Framework

The APM delivery framework is developed to support an organisation's business objectives; it "emphasises execution and it is explanatory rather than deterministic" (Highsmith, 2010:82). In order to achieve business objectives, the framework must (Highsmith, 2010:81):

- support a disciplined and self-organising project team;
- promote consistency and reliability as far as possible, given the level of uncertainty there might exist in the project;
- support an *adapt, envision and explore* culture;
- incorporate practices that support each project phase;
- incorporate learning;
- be adaptable;
- support a transparent view into the process; and
- supply checkpoints for management for evaluation.

The APM delivery framework as presented in Figure 3.7 demonstrates the five phases of APM. The phases must be seen as one phase flowing into the next and not like encapsulated separated phases – "the APM terms were selected to imply iterative evolution" (Highsmith, 2010:87).



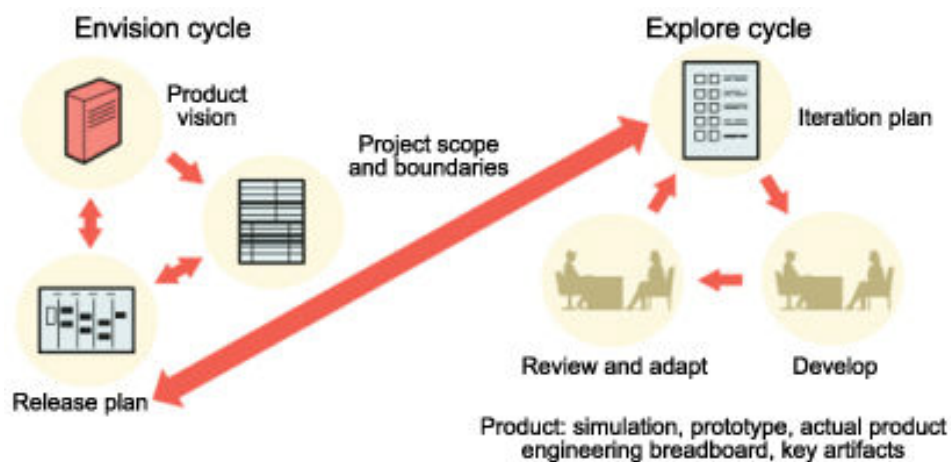
Source: Highsmith (2010:81)

Figure 3.7: The APM delivery framework

Agile Project Management consists of five phases, namely: envision, speculate, explore, adapt and close. The APM delivery framework moves away from the traditional phase names such as initiate, plan, execute, develop, test, etc. This has a great significance, as it indicates that APM embraces change and changing as often as required. The traditional initiation phase is replaced by the envision phase to signify the importance of a vision. The customary planning phase, which is normally “associated with prediction and relative certainty”, is replaced by the speculate phase in order to highlight the uncertainty of the future and the need for adaptability, as the end result cannot be predicted (Highsmith, 2010:82). “Many traditional project managers faced with uncertainty try to plan the uncertainty away. We must learn to speculate and adapt rather than plan and build.” (Highsmith, 2010:82). The build, design and test phases are replaced by the explore phase, which is a non-linear and non-waterfall model that focuses on iterative development. In this phase, the uncertainties that are identified in the speculate phase are analysed, solved and explored. The adapt phase ensures that the team members remain focused on the vision and adapt to the current environment. During the last phase, close-out, the project is handed over and knowledge is transferred.

3.5.3.1 Envision

Any good CEO, project manager or other leader emphasises the importance of a vision. For this reason, APM defines the project’s vision in the first phase to ensure that everyone involved understands what has to be achieved by when in order to determine whether the project has been delivered successfully. The cycle of the envision phase as presented in Figure 3.8 functions closely with the cycle of the explore phase.



Source: Highsmith (2010:204)

Figure 3.8: The envision and explore cycles

During the envision phase, the project scope, project community and stakeholders, product vision, and the manner in which the team works together are determined. Firstly, the team must envision what must be delivered. In order to do this, the project objectives, constraints, boundaries and vision must be defined and understood. Secondly, the team has to identify which stakeholders and members of the business community will be involved. Lastly, the team must decide on the manner in which they will work together in order to deliver the expected and defined vision of the project. Release planning (as seen in Figures 3.7 and 3.8) is mostly done during the speculate phase. The following questions are answered in this phase (Highsmith, 2010:94):

- What is the client's product vision?
- What are the key capabilities required of the product?
- What are the project's business objectives?
- What are the project's quality objectives?
- What are the project's constraints (scope, schedule and cost)?
- Who are the right participants to include in the project community?
- How will the team deliver the product (approach)?

In smaller IT projects, the envision and speculate phases can even be completed during the project kick-off meeting when the whole project team is present. In larger IT projects, some project activities, such as procurement and training, can be included in the first iteration (also called iteration 0) of the speculate phase. According to Highsmith (2010:95), it is important to remember that:

- The team members should constantly determine the bare essentials of the process and documentation needed.
- All the practices related to the manner in which a team delivers are tailored and adapted to improve performance as the project progresses.
- The project community will evolve its practices where everyone works together.

Creating a vision ensures that everyone associated with the project works towards the same goal as a whole. It is also important that each team member understand his/her role in each of the defined project goals, otherwise they will neglect their responsibilities and this may cause the project to fail.

The envision phase has four categories of practices:

1. "Product vision:
 - product vision box and elevator test statement;
 - product skeleton architecture and guiding principles;

2. Project objectives and constraints:
 - project data sheet;
3. Project community:
 - obtaining the right people;
 - participant identification;
 - client team–development team interface;
4. Approach:
 - process and practice tailoring.”

3.5.3.2 Speculate

Predicting what will happen in the future during IT projects may turn out to be a costly exercise because of a changing project environment. As soon as clients see what is possible, they always desire more. For this reason, it is better to speculate than to plan because the future is unknown, bearing in mind that speculation is based on what is already known. The term “speculation” may appear to imply reckless risk taking, while it actually means conjecturing project expectations based on incomplete facts and information (Highsmith, 2010:83). Agile Project Management is more than just planning and doing – it is about creating a vision and exploring it because only some information is available and this must be examined to “determine our course of action in the next iteration” (Highsmith, 2010:129).

During the speculate phase, planning is not ignored, “but speculating is a more appropriate term as uncertainty exists” (Highsmith, 2010:129). Planning in the business world is an overused word and implies that planning can overcome uncertainty. Planning cannot overcome uncertainty however; rather, project managers and teams can speculate and adapt according to their project’s circumstances and expectations.

The speculate phase entails (Highsmith, 2010:84):

- collecting the primary client requirements for the product;
- defining the amount of work as a backlog of product features;
- developing an iterative release plan that is feature based;
- integrating risk mitigation strategies into the plan; and
- estimating the cost of the project and collecting other important administrative information.

The speculate phase produces a release plan, which is not based on activities like traditional IT projects, but on stories, like XP. During this phase, the product structure, backlog of stories and capabilities, and the release plan are defined, understood and created (Highsmith, 2010:129). Feature (time-boxed) planning and development are executed by keeping the client involved by

explaining the whole process of delivering the product(s) and project. Change is also more easily managed when clients are involved because they understand the difficulties and constraints the project team is facing. Features, which take on the form of story cards, are used as a communication technique for shared understanding between the development team and the client. The features are then broken down for development and implementation.

During the envision phase, a product breakdown structure is created by identifying and listing the features required to deliver the product. During the speculate phase, the list of features is expanded by creating story card(s) for each feature, where the front of the story card contains requirements for planning purposes and the back contains technical information that assists team members in estimating the time and effort required to complete the client's expectations. The story cards are then analysed in more detail to be developed and tested within a specific planned iteration during the explore phase.

When listing and expanding the features, it is important to remember the vision, aims and objectives of the project, which is the reason the release plan is so important. The release plan "represents a roadmap of how the team intends to achieve the product vision within the project objectives and constraints" (Highsmith, 2010:142).

Agile project speculation assists the project team in (Highsmith, 2010:131):

- determining the manner in which the product and its features will evolve in the current release;
- balancing anticipation with adaption as the project unfolds;
- focusing on the highest-value features early in the project;
- thinking about the business goals, project objectives and client expectations;
- providing necessary cost and schedule information to management;
- establishing priorities for trade-off decisions;
- coordinating interrelated activities and features across teams;
- considering alternatives and adaptive actions; and
- providing a base-line for analysing events that occur during the project.

3.5.3.3 Explore

This phase cannot be likened to an explorer discovering the unknown, because the vision is defined of what the project team must achieve without reckless risks. During the explore phase, stories (features) are planned, developed, tested and delivered in short iterations while the objective is to constantly reduce the uncertainty and risk of the project.

In Figure 3.8, the release plan executed during the envision phase links to the iteration plan of the explore cycle. The figure also represents three major activities with sub-activities that work in conjunction with the following activity areas in this phase (Highsmith, 2010:84):

- planning and delivering stories by managing the workload and using suitable risk mitigation strategies and technical practices;
- facilitating and creating a self-organised and collaborative project environment; and
- managing communication and interaction amongst stakeholders, clients and product management.

The first major activity is integration planning and monitoring, which entails iteration planning, workload management, and monitoring the iteration progress – which is managed by the iterations manager. Iteration planning is done by the team by using each story card in the release plan to plan for the first or next iteration(s). The tasks to complete the listed story cards for the iteration that must be executed are listed, after which the team re-estimates the work effort and reprioritises the story cards where necessary. The workload is managed by each team member, as he/she is responsible for delivering the stories when the iteration is complete. The progress of a specific iteration is monitored on a daily basis using the daily stand-up meetings, as is the case with Scrum.

The second major activity is technical practices, which involves simple design, continuous integration, rigorous automated testing, and opportunistic refactoring of which most are specific to the product engineering domain – in order to keep costs of change low and to deliver a product of high quality. The four technical practices work in concert with each other and are crucial for the effective adaptability of a project and the project team.

The technical department is brought onto a project when a project team is under pressure to complete stories (deliverables). This reduces the speed of development and it negatively affects the team's ability to deliver. The aim of simple design is to keep the team focused on what they know, instead of anticipating what they do not know. There are two fundamental approaches to managing change (Highsmith, 2010:218):

- *Anticipation*: Plan for the future and predict what kinds of changes are probable.
- *Adaptation*: Wait until requirements or development issues have been defined and build them into the product.

Continuous integration at the beginning stages of the project and during development aims to ensure that the product features (stories) are integrated as a whole to reduce the burden of testing and the high cost of misalignment. "The less frequent the iterations, the more

susceptible the development effort will be to major problems late in the process and the more difficult, and expensive, it will be to find and fix them.” (Highsmith, 2010:220). The aim of rigorous automated testing is to ensure that the quality of the product constantly remains high during the development process. A team will be more effective if it has tested and running features for each iteration. During opportunistic refactoring, the project team focuses on continual and constant improvement, as the product is designed to make it more flexible and adaptable in order for the team to deliver features on a continual basis.

The third major activity relates to the project community. It entails coaching and team development, participatory decision-making, collaboration and coordination. During coaching and team-building, the team explores, experiments and learns from the mistakes they make. The agile project manager’s aim is to build a high performance and focused team by realising each team member’s individual talents. The aim of participatory decision-making is to keep the team members and stakeholders involved and to provide project practices to assist in decision-making as the project progresses. In order ensure that collaboration and coordination between stakeholders occurs during a project, practices such as stand-up meetings (Scrum), stakeholder coordination and daily interaction with the project team are applied.

3.5.3.4 Adapt

Adapt or die is a common phrase, meaning that if anything does not adapt to its environment it will not survive; this principle applies to IT projects. According the Highsmith (2010:85) “adapt implies modification or change rather than success or failure”. In order to adapt, there must be an understanding of the risks, changing requirements, project processes and the market. Every agile project team must evaluate and adapt in the following areas (Highsmith, 2010:235):

- product value;
- product quality;
- team performance; and
- project status.

The term “corrective action” is commonly used between teams and project managers. This term implies that the team has underperformed or made an error and that action must be taken in order to re-align the project with the project plan, which in turn implies that the project plan is always correct. Agile Project Management replaces the term with “adaptive action”. This term implies that the team responds to events or incidents (rather than trying to correct) because the project plan was originally developed based on assumptions and speculations. This adheres to the Agile Software Development Manifesto’s value of responding to change rather than following a plan. The team has to answer four difficult questions, as it is more difficult to adapt to

events than correct a project plan (Highsmith, 2009:254):

- Is value, in the form of a releasable product, being delivered?
- Is the quality goal of building a reliable, adaptable product being met?
- Is the project progressing satisfactorily within acceptable constraints?
- Is the team adapting effectively to changes imposed by management, clients or technology?

In the adapt phase, the achievements of an iteration and project progress are reviewed by the project team and stakeholders, such as the client. The feedback provided gives information on the actual versus the predicted time, cost and quality estimates, and up-to-date project progress and status information. The lessons learnt and output are used during the next iteration's re-planning effort. Conducting review and adaptive action sessions at the end of an iteration is very important, as this allows the team to reflect, learn and adapt where necessary, and to create a change in pace and urgency in executing the project iterations.

After creating a vision in the envision phase, the speculate, explore and adapt phases loop for each iteration to refine the product in development. There might be instances in which the vision of a project in the envision phase must be reviewed, as new and changing requirements or information is gathered that might affect the course the project should take.

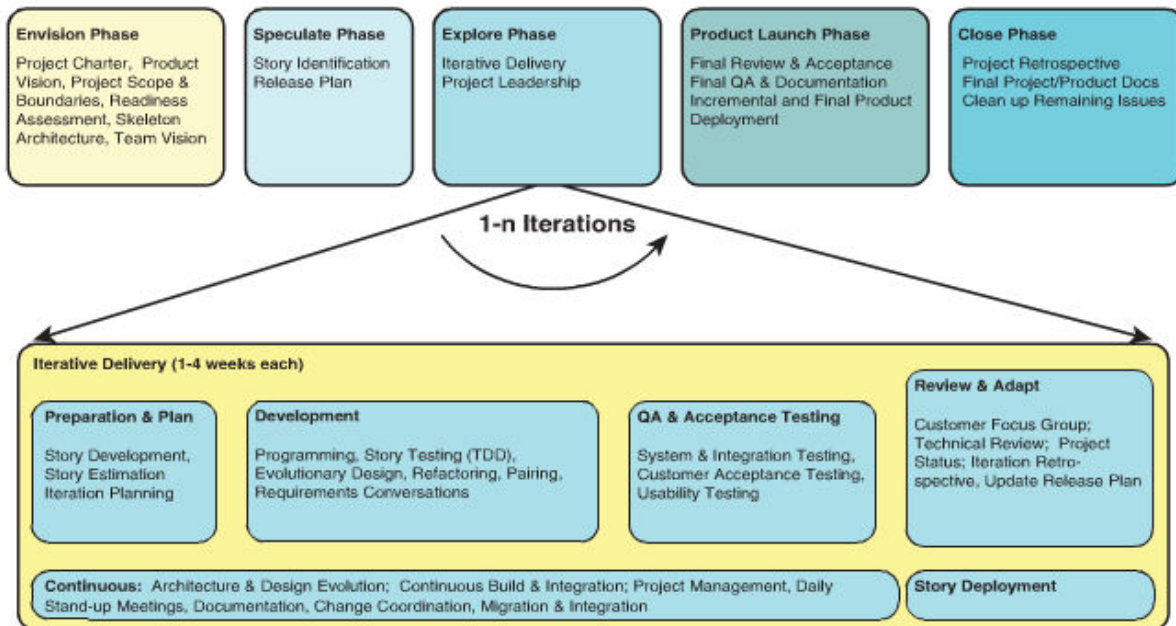
3.5.3.5 Close

Closing out a project is often seen as a time-consuming phase with not much value to the client, although it is actually just as important as the other phases of the project. By compiling a close-out report, the project team summarises what has been completed successfully and finally agrees that the expectations of the client have been met. Even if a project causes the initiation of another project, it is still important to obtain sign-off and close-out of what has been successfully completed to ensure that the client does not later complain that what has been completed is actually not what they required or that the project is not working because the previous project's expectations were not satisfied. The main aim of this phase, and the small close-outs of each iteration, is the lessons learnt, which must be used for the planning and execution of the next iteration or project. When closing out a project, it is a good idea to have a celebration and short award ceremony.

Concluding remarks

The APM framework presented by Highsmith (2010:81) is not a complete life cycle. The APM is not an internationally accredited PMM like PMBOK and PRINCE2, but it demonstrates characteristics of a new way of managing projects and delivering them successfully, which

cannot be ignored. Using the explanation of the APM delivery framework above, an expanded APM delivery framework is presented in Figure 3.9 as the APM life cycle.



Source: Highsmith (2010:89)

Figure 3.9: The APM life cycle

In this expanded life cycle, the APM delivery framework becomes a hybrid PMM. The explore phase is broken into detailed iterations, showing what needs to be completed during each iteration. The figure also shows practices that must be adhered to during every phase.

3.5.4 Evaluation of Agile Project Management

Agile Project Management is a very new in relation to PRINCE2 and PMBOK, which have been developed and improved over the last 20 years. As APM is so new, it is difficult to research where it has been applied in practice successfully, what the advantages and disadvantage are, or whether it can be integrated with other methodologies. In addition, Highsmith (2010) admits that the APM is not a complete life cycle and still needs to go through necessary revisions for improvement.

Application area

According to Puri (2009:43), there is an urgent need for agile projects to be managed because there is a perception that when one uses ASDMs, one does not require project management. In today's fast-paced and ever-changing business environment that requires a PMM that can adapt, business and project managers have begun to harness and apply APM for much more than just software development (Moore, 2009:148–149). Instead of applying traditional PMMs,

new PMM approaches should include the four fundamental values of APM and ASDMs (Rico, 2010:41):

- client collaboration;
- iterative development;
- self-organising teams; and
- adaptability to change”.

Agile Project Management presents the opportunity to deliver different types and different sized projects across different industries because of its adaptive nature. It is praised by many CEOs, directors and academics from companies and universities across the world (Highsmith, 2010), which demonstrates that this PMM has something of value to offer. There are few sources that report on the effectiveness of APM; however, APM is one of the latest PMMs and this may explain the absence of information in this regard.

Advantages

The benefit of APM is that it can be used to manage dynamic projects with a high level of uncertainty, urgency and complexity more effectively (Hass, 2009:9). In these types of projects, there is a need to combine the roles of a business analyst and a project manager into one role (Wysocki, 2011:126). Agile Project Management offers the following advantages (Virine & Trumper, 2008:156):

- a creative environment is created that is free of frustrated developers;
- to seek and determine preventative and corrective actions for project risks; and
- to manage adaptively and to review every decision made.

In the midst of organisational chaos, APM has the ability to survive by adapting continually to the changing project environment. Agile Project Management accepts that change is inevitable and that change must not be avoided but managed and controlled, while traditional PMMs aim to minimise change by extensively planning and documenting as much as possible before development is commenced. This PMM is lean, flexible, easy to use, and individuals do not have to undergo intensive training to utilise the PMM correctly. It is developed for use at a technical level, rather than a management level. Furthermore, APM is different from other PMMs, as it assigns iteration managers to assist project managers by managing the internal team members and the individual iterations.

The three core values explained by Highsmith (2010:14–17) revolutionise the approach towards project management, according to which the project manager has to adapt, lead the team and deliver value, instead of conforming to the plan, managing tasks and meeting constraints, the

aspects focused on by all traditional PMMs. The intent of the simple design is to keep the team focused on what they know, instead of anticipating what they do not know. The vision is defined and refined regularly as the project progresses.

Agile Project Management is distinguished by speculation and iterative development – instead of planning. The project team speculates based on what they already know, but planning is not ignored. It just accepts that the future is unknown, while other PMMs seek to prevent uncertainty through planning.

Agile Project Management is the only PMM studied that incorporates short daily meetings, like Scrum, in order to ensure that there is regular stakeholder and team member communication and problem resolution. Regular and continuous testing is embedded throughout the PMM to ensure constant, high-quality product delivery.

Disadvantages

Agile Project Management is built on the same foundations of ASDMs, which exposes it to the same climate of criticism and appraisal. Many of APM's components are similar to those of XP and Scrum, which makes it vulnerable in the sense that if someone or some organisation does not like XP or Scrum as ASDMs they may neglect or drop APM as a whole.

It still needs to undergo in-depth evaluation and an improvement process in order to make it a PMM that will be accepted as a global PMM, like PMBOK and PRINCE2. It lacks certain management areas that are essential for good project management, such as human resource management, issue management and procurement management.

One focus of APM is to keep documentation to a minimum, which might cause problems, especially in a high-tech project environment in which documentation is viewed as the main reference source for any information or decisions to be made. However, if documentation is not viewed as important, this may cause confusion and misalignment owing to team members not being sufficiently guided by documentation. For this reason, there needs to be a balance between the importance of documentation and the volume of documentation. Documentation must still be viewed as very important because it is used as the benchmarking mechanism in many organisations to govern the execution of a project. The only documentation that must be avoided is documentation that will never be read.

Agile Project Management is developed to be executed at a technical level, the project manager must thus be careful not to focus only on individual iterations and must bear the project's vision

in mind continuously. It does not give much attention to setting up a business case, which makes evaluation at project close-out a problem. When the project is evaluated to determine whether it was successful, the business case should be used as a base-line to check whether the client's expectations have been satisfied. If the business case is included in the vision, then benefit realisation might be more accurate and effective.

Integration with other methodologies

Agile Project Management is a “rational concept in project management and it helps to mitigate some common biases and mental traps common in waterfall approaches” (Virine & Trumper, 2008:156). According to Dingsoyr *et al.* (2010:4), it is possible to integrate APM with “overall traditional principles, such as the stage-gate project management model”. The latter is a model according to which a project needs to progress through a series of stages and gates to reach completion.

Agile Project Management is the only PMM that provides an enterprise framework that assists organisations to develop hybrid ASDMs to adapt to each project and organisational environment. It is in essence a hybrid SDM itself that largely consists of Scrum, XP, ASD and the Agile Software Development Manifesto (Beck *et al.*, 2001) components. This introduces the possibility of APM to be combined with other methodologies found in practice to improve the approach.

3.6 Comparison of PMBOK, PRINCE2 and Agile Project Management

Comparing the different PMMs with each other in order to highlight certain similarities and differences can be difficult because the structure and design of each studied PMM differs considerably. All the PMMs address the well-known triple constraints of time, cost and scope. A common characteristic of these PMMs is that all projects are finite, meaning that they differentiate a clear start and close-out of a project. All the PMMs emphasise that a project cannot be initiated if the scope, time and cost constraints, and deliverable expectations are not defined. This is a positive attribute of all the PMMs, as it is important to ensure that a project is profitable and adds value before it is initiated. Furthermore, these PMMs view the management of relationships amongst the project team, project manager, project board, client, users, other stakeholders and the rest of the project and business environment as very important – this emphasises that communication plays an integral role in delivering satisfactory IT projects. All the PMMs also explain the manner in which they can fit into the business enterprise and they view the project and business community as important, although only PRINCE2 and APM allow PMM adaptation to its environment to increase the likelihood of delivering IT projects in unique and changing business and project environments.

The underlying approach differences amongst these PMMs is that PMBOK offers a framework-oriented approach to managing projects using knowledge areas, while PRINCE2 adopts a process-oriented approach that consists of a logical sequence of flexible steps, and APM provides a flexible approach. PRINCE2 is pragmatic and more of a project implementation methodology, rather than a whole PMM (Wideman, 2002:4; Yeong, 2009:8), while PMBOK is more comprehensive (Yeong, 2009:8). PMBOK and PRINCE2 are both generic; thus, that they can be applied to any type of project (Nelson & Nelson, 2006:219).

PMBOK was initially developed to manage any type of project. PRINCE2 was originally designed for IT projects although it was found to be successfully for non-IT projects as well. Agile Project Management is a flexible and adaptable approach to project management, in which deliverables are broken up into increments and iterations to be delivered in a constantly changing project environment and was found to be very successful in IT projects. The success of using APM in other project types must still be determined, as it is still a very young PMM in the marketplace.

PMBOK is used mainly in the USA, from where it originated, while PRINCE2 is widely used in the UK because it was originally designed by the UK government. Both PMBOK and PRINCE2 are globally accepted PMMs, of which both have professional certifications which can be obtained in different languages. Organisations are increasingly starting to adopt APM because it has ASDM characteristics and ASDMs were found to be successful in delivering IT projects within changing project environments as explained in Chapter 2.

A comparison was drawn amongst the differences and similarities of the PMMs with regard to PMBOK's five process groups, PRINCE2's seven processes and APM's five phases. This is shown in Table 3.5. Following this comparison, a more detailed comparison is offered below.

Table 3.5: Mapped phases of PMM life cycles

PMBOK	PRINCE2	APM
Initiation	Starting up a project Directing a project Managing a stage boundary	Envision
Planning	Initiating a project Managing a stage boundary Managing product delivery	Speculate
Execution	Controlling a stage Managing product delivery Directing a project	Explore
Monitor and control	Controlling a stage	Adapt
Close	Managing a stage boundary Closing a project	Close

Source: Adapted from Project Plus (2008); Sliger (2006:5); Yeong (2009:9–10)

PMBOK and PRINCE2 have process groups (non-isolated) that are applied to each phase or stage of a project or to a project as a whole, while APM use phases and breaks down the project into multiple iterations in the *explore* phase. With regard to PRINCE2's DP process group, the project board is constantly involved in the project, just like user and project board involvement in APM. In PRINCE2's IP process group, quality management, communications management, planning and risk management are viewed as very important, just like PMBOK. PRINCE2 is the only PMM that addresses configuration management as part of the IP process. PRINCE2 does not address procurement management, contract administration, human resources management, communication management or cost control, while these areas are addressed by PMBOK (Yeong, 2009:1). In some instances, PRINCE2 falls "short of providing project teams with the knowledge of how to complete certain activities", but PMBOK provides guidelines on this (Bell, 2009:5).

With regard to the PRINCE2 themes, like PMBOK, PRINCE2 views the business case as a very important document that must be completed during the initial stages of a project and progress must be measured against the objectives and constraints as explained in the business case to ensure that what is promised to the client is delivered (OGC, 2009; Portman, 2009; Bentley, 2005). Agile Project Management merely mentions the importance of a business case in a project. It does not focus on documentation but ensures that clients are always involved and adapts to change within the project environment. PMBOK stipulates requirements and seeks to ensure that the budget and project schedule are planned and calculated correctly, while APM emphasises that the project environment and scope constantly change, which causes the schedule and cost not to be fixed (Sliger, 2006:4).

PRINCE2 and APM have the flexibility to adapt and to be tailored to the project environment. They also manage on a stage-by-stage or increment-by-increment basis by focusing on delivering valuable products and not completing activities (OGC, 2009:5, 12, 150; Portman, 2009:8; Bentley, 2005:13; Highsmith, 2010:16), while PMBOK is not process-orientated and the framework provided is not flexible. Projects should not constantly measure team members performance against the project plan, otherwise adaptability will suffer (Highsmith, 2010:132). If responsiveness to client expectations, innovation, flexibility and adaptability is required, then the project board and project manager must not reprimand the team members, but instead reward them for their responsiveness to the changes that occurred (Highsmith, 2010:132).

Like APM, PRINCE2 manages every stage boundary (SB process) to ensure that the project is well controlled by managing any changes, risk or issues that might occur and thereby deliver what was promised when the project was initiated. Agile Project Management and PRINCE2

allow the review of the stage or increment completed in order to reshape the next stage or increment by applying the lessons learnt in the previous stage or increment. Agile Project Management, like PRINCE2, provides a controlled change-management approach (Bentley, 2005:5) that is scalable, flexible and adaptable (Richards, 2007:8; OGC, 2009:5, 12–14). Because both PMMs divide the project into increments or manageable units, they provide effective change control (sixth PRINCE2 theme), progress monitoring and flexible decision points (Richards, 2007:8).

One of the main values of APM is that it adapts to change rather than conforming to plans. This can be very risky if not managed correctly because if a person focuses on adapting to change only and does not adhere to the constraints of the project, the likelihood of running over budget is increased. The project plan must be viewed as a guide and thus meant to be flexible (Highsmith, 2010:63). The project manager must ensure that he/she executes the necessary change-control procedures to ensure that the project remains well controlled. Instead of using the term “corrective action”, which implies that the team has made a mistake and that an action is required to correct this mistake to ensure the project remains on target, APM uses the term “adaptive action”, which implies that the team reacts to events or incidents because predictive project plan are based on assumptions and speculations. Agile Project Management terms the person responsible for a stage(s) or increment(s) an “iteration manager”, while PRINCE2 terms the same person a “product manager”.

Agile Project Management is built on the principle that change is inevitable and that a project needs to adapt in order to deliver value, while PRINCE2 and PMBOK focus on planning what must be done and executing it according to the plan, allowing change only when and where necessary. PMBOK is easy to understand because there is a good understanding of what is required in the knowledge areas, but it does not provide possible templates to execute the requirements – the same is true for APM. PRINCE2 however provides the necessary information, documentation and templates or frameworks to complete business cases, feasibility studies, project charters, etc.

Agile Project Management is the only PMM that mentions other management areas, such as contract and documentation management, that can be included as part of the PMM (Highsmith, 2010:79). Some areas not addressed by any of the PMMs are knowledge management and legislation and regulations, which are areas essential to the success of IT projects. Business politics and the human factor are also not considered by any of the PMMs. Humans are unique beings and each individual behaves differently, especially when under stress. An individual’s behaviour in certain circumstances cannot be predicted and this unpredictability may cause the delay or even failure of the project.

In order to compare the PMMs, the knowledge areas of PMBOK are used as the basis and PRINCE2 and APM are compared with this. This comparison is given in Table 3.6.

Table 3.6: PRINCE2 and APM component comparison with reference to PMBOK's knowledge areas

PMBOK	PRINCE2		APM
Integration	Principles:	7 principles	• Continuous integration • Iterative envisioning and planning • Release plans and iteration plans • Technical communication solutions for information sharing • Daily stakeholder and team integrated change control
	Themes:	Organisation, plans, change, progress	
	Processes:	7 processes	
Scope	Principles:	7 principles	• Expect scope change • Fixed cost and schedule • Part of release plan and iteration plan
	Themes:	Business case, plans, change	
	Processes:	7 processes	
Time	Principles:	Continued business justification, learn from experience, manage by stages, tailor to the project environment	• Fixed cost and schedule • Part of iteration planning
	Themes:	Plans, change, progress	
	Processes:	7 processes	
Cost	Principles:	Continued business justification, learn from experience, tailor to the project environment	• Fixed cost and schedule • Integrated change control
	Themes:	Plans, change, progress	
	Processes:	7 processes	
Quality	Principles:	Continued business justification, learn from experience, focus on products, tailor to the project environment	• Fixed cost and schedule • Optimised value • Part of analysis and design • Involved in decision-making through entire life cycle • Test-driven development • Daily team involvement and participatory decision-making • Daily quality and quality assurance
	Themes:	Quality, plans, change	
	Processes:	7 processes	
Risk	Principles:	Learn from experience, manage by exception, tailor to the project environment	• Daily risk identification • Daily mitigation strategies • Team and stakeholder involvement and participatory decision-making
	Themes:	Risk, change, progress	
	Processes:	7 processes	
Communication	Principles:	Learn from experience, tailor to the project environment	• Technical communication solutions for information sharing • Daily stand-up (Scrum) meetings • Daily team and stakeholder involvement
	Themes:	Risk, change, progress	
	Processes:	7 processes	
Human resources	Principles:	Learn from experience, define roles and responsibilities, tailor to the project environment	• Establish cross-functional team • Self-organising team • "Servant leadership" mentality
	Themes:	Organisation, plans	
	Processes:	7 processes	
procurement	Principles:	Continue business justification, learn from experience, focus on products, tailor to the project environment	• APM does not address procurement management explicitly, although some aspects of procurement management are integrated throughout the PMM.
	Themes:	Business case, plans, change, progress	
	Processes:	Initiating a project, managing product delivery	

Source: Adapted from Project Plus (2008); Yeong (2009:9–10)

In order to select the most appropriate PMM, each of these PMMs must be understood and their applicability to a specific project environment must be studied. There is no one particular PMM suitable for all projects, as each project is unique and must be managed in such a way that it is delivered successfully. Building a hybrid that contains the best practices of each of the PMMs will be the main focus of Chapter 4. Many of the strengths of these PMMs can be combined to build a hybrid methodology with a better ability to deliver IT projects in different environments, than the general PRINCE2 and PMBOK approaches widely used today.

3.7 Summary

In this chapter, the term “project management” was defined, after which three PMMs was investigated. The first PMM described was the fourth edition of PMBOK, an internationally recognised standard and project management guide. The second PMM described was the fifth edition of PRINCE2, which was originally developed as a UK Government standard for IT projects. The third PMM described was the relatively new APM, which has the unique ability of adaptability to a changing project environment. Each PMM was evaluated in relation to their area of application, advantages, disadvantages, and integration with other methodologies. Lastly, the similarities and differences of these PMMs were investigated.

The evidence provided in Sections 3.3.4, 3.4.4 and 3.5.5 supports the conclusion that PMMs can deliver IT projects within cost, time, quality and scope constraints if executed correctly. Although PMMs can deliver IT projects successfully, they also have some shortcomings and disadvantages, which demonstrates that PMMs alone are not able to provide assurance that a project will be delivered *successfully in a constantly changing environment*. For this reason, in this study the strengths of ASDMs and PMMs were combined and the identified weaknesses or flaws of each were addressed by developing a hybrid APMM that has the ability to deliver IT projects in a constantly changing project and business environment (to be discussed in Chapter 5). The research methodology followed in developing this hybrid APMM will form the main focus of Chapter 4.

CHAPTER 4

RESEARCH DESIGN

4.1 Introduction

In this chapter, the way in which the research design was applied will be explained. Firstly, the positivistic, interpretivistic, critical social theory and mixed methods paradigms will be explained by looking at the definition, history, characteristics and criticism of each. Design science as research method will then be explained, after which the research plan will be presented using the design science cycle as foundation.

The possibility of merging ASDMs (studied in Chapter 2) and PMMs (studied in Chapter 3) will be discussed, by combining positivistic design science, positivistic surveying and interpretive case study research as a mixed methods research approach to creating versions 0, 1 and 2 of the hybrid APMM. The manner in which the different versions of the hybrid APMM were developed will be explained in the research plan, in which the mixed methods paradigm (interpretivism and positivism), research methods (design science, case study and survey), data-collection techniques (interviews and questionnaires), and data-analysis techniques (cross-case and statistical) will be detailed. Each paradigm, research method, data-collection technique and data-analysis technique will be explained theoretically before detailing its practical application in this study.

4.2 Research paradigm

According to Oates (2006:282), a paradigm is a “set of shared assumptions or ways of thinking about some aspect of the world”, in which the major concern is a shared view of the manner in which research is executed and knowledge is gained in different communities. Creswell (1998:74) defines a “paradigm” as the “basic set of beliefs or assumptions” used to guide a researcher’s perspective. Research paradigms have different epistemologies (manner in which knowledge can be obtained about the nature of the research setting) and ontologies (different views and perceptions about the nature of the research setting; Oates, 2006:282).

In this section, the positivistic, interpretivistic, critical social theory and mixed methods research paradigms will be described, first by defining each of them. A short history of each paradigm will be provided after which the general characteristics and criticism (strengths and weaknesses) of each paradigm will be discussed.

4.2.1 Positivism

Positivism focuses only on knowledge obtained from facts, experience and evidence. Positivists aim to generalise to a whole population, based on the analytical findings of the sample under investigation. The Oxford Dictionary defines “logical positivism” as “a philosophical system recognising only that which can be scientifically verified or which is capable of logical or mathematical proof, and therefore rejecting metaphysics and theism” (Oxford University Press, 2011). Positivism assumes that the research environment is studied objectively, and that the world or research setting is not random but usual and ordered (Oates, 2006:283). The following perspectives characterise positivism (Giddens, 1974:2):

- There is a fundamental difference between values and facts. Values belong to an entirely different level of discourse outside the realm of science, while science deals with facts.
- Natural science methods can be applied in social sciences, and human and natural sciences share the same methodological and logical foundations.
- Science consists of a framework by which any form of knowledge can be established.
- What is available to the senses forms reality.

The positivistic concept was developed by the French philosopher Auguste Comte (1798–1857) in the early nineteenth century. Comte (1896:17–18) claimed that a society’s salvation was dependent upon scientific knowledge and that civilization (social behaviour, values, beliefs, etc.) could be studied using natural science research methods.

Émile Durkheim (1858–1917) refined the basic aspects of Comte’s positivistic philosophy, although he rejected much of the detail. Durkheim (1985:21) recognised that society was a moral phenomenon that must be studied in groups with the same manner of thinking of a chemist, physiologist or physicist when seeking new scientific findings. There are other positivist philosophers, such as Karl Popper (1902–1994), Wilhelm Scherer (1841–1886) and Dimitri Pisarev (1840–1868) with their own views regarding positivism.

Positivism has the following characteristics (Oates, 2006:286):

- *World, including the social world, exists independently from our mind:* Social and physical world exist that must be studied, analysed, measured, captured and understood.
- *Universal law:* Generalisations are the focus of the researcher, where indisputable facts, patterns, or universal laws are proven and presented to be true regardless of the research setting.
- *Measure and model:* Working models are produced by the researcher that explore the world by observing, analysing and measuring its nature.
- *Researcher’s objectivity:* The researcher is an objective observer.

- *Quantitative data analysis*: Statistical analysis and mathematical proofs are the preferred data-analysis methods.
- *Testing of hypothesis*: Research is based on the empirical testing of hypotheses and theories, which can be accepted or refuted based on the results of the testing.

Positivism is the oldest research paradigm of the four paradigms studied, and is structured for the original purpose of studying the natural world, such as magnetism, atoms and gravity (Oates, 2006:288). Because positivism has been in existence for hundreds of years, it can be assumed that there will be much criticism of it, although it is normally used as the benchmarking paradigm for other research paradigms. Criticism of positivism includes (Oates, 2006:288–289):

- Reductionism is not possible when studying the social world, such as people, the groups they form, the meanings they create and the cultures they form.
- It is sometimes impossible to break something complex down into smaller and simpler researchable parts (reductionism).
- Repetition of the study is not always possible.
- Generalisation is often not desired.
- Everybody interprets the world differently.
- Laws and patterns are observable in the social world and they are the creation of people.

Owing to the shortcoming identified of positivism not being able to be applied meaningfully to the study of people and their interpretations of the world, alternative research paradigms were developed, such as interpretivism and mixed methods. Interpretivism will be discussed in the next section.

4.2.2 Interpretivism

Interpretivism, also known as antipositivism, is a social science philosophy that is interested in the understanding of human action and behaviour from the actor's perspective. Interpretivists do not identify events and actions as observations of external reality but within human society (Hughes, 1990:90). Interpretive research "is concerned with understanding the social context of an information system: the social processes by which it is developed and construed by people and through which it influences, and is influenced by, its social setting" (Oates, 2006:292).

Antipositivism evolved in the late 1800s, when sociological positivism began to be questioned by German philosophers, Wilhelm Dilthey (1833–1911) and Heinrich Rickert (1863–1936). They argued that the world of society is different from the world of nature, as human societies have cultures, which includes unique attributes such as values, norms, meanings and rules. According to Hughes (1990:90), Dilthey explained that history represents much more than just a

list of consecutive events, it signifies “the spirituality of social life as expressed in social institutions, law, literature, government, morality, values and more”. Dilthey (1989:66) explained that human behaviour cannot just be understood by reason alone, it also requires spiritual understanding and ability.

Antipositivism was taken further by the German sociologist Karl Emil Maximilian Weber (1864–1920) and his associate Georg Simmel (1858–1918). According to Ashley and Orenstein (2005:239–241), Weber sought to determine relationships that were not as generalisable, ahistorical, or invariant as those pursued by natural scientists, and he furthermore argued that sociology may be loosely described as a science because it is able to recognise causal relationships of human social behaviour – particularly amongst *ideal types*, or hypothetical simplifications of complex social phenomena.

Interpretivists state that positivist attempts to measure human behaviour are inadequate because they exclude the constructed and inter-subjective nature of the social world. Interpretivism recognises, explores and describes all the mutually dependent and related factors within the social environment and has the following characteristics (Oates, 2006:292–293):

- *Numerous subjective realities*: There is more than one version of the truth.
- *Dynamic, socially constructed meaning*: Whatever a person or group views as reality can only be accessed and conveyed to other people through communication, understanding and meaning.
- *Researcher reflexivity*: The researcher must be self-reflective, as his/her own beliefs, line of thought, meaning, actions, values and assumptions will affect the research setting, as he/she will unavoidably shape and direct the research process.
- *People are studied in their natural social setting*: People are studied from the participants’ perspective without the researcher manipulating or imposing his/her experience or understanding of the environment under investigation. The focus of interpretivism is the understanding of individuals and their perspectives in a real world, not an artificial one.
- *Qualitative data analysis*: There is a preference for gathering and analysing qualitative data (though interviews and cross-case analysis).
- *Multiple interpretations*: The researcher expects that there will not just be one answer or explanation to the research question. More than one explanation or answer will be provided and the one that appears to have more evidence for its assertion will be discussed. Every participant is unique and interprets and describes his/her environment and narrative of events differently, which is the reason that the researcher must be sensitive to contradicting interpretations and he/she must be able to identify multiple interpretations of a specific situation (Klein & Myers, 1999:72).

Interpretive research is conducted by applying the following principles (Klein & Myers, 1999:72): *Hermeneutic cycle*: This principle of human understanding is fundamental to all the principles. By allowing interaction between the mutually dependent parts and the whole, the form of the parts will facilitate human understanding. The parts are the sole understanding of the participants and researcher of the problem situation. The consolidated meaning created from the interaction between the researcher's and participants' understanding of the problem situation forms the whole.

- *Contextualisation*: In order for the audience to understand the historical context and the circumstances of the current problem situation under investigation, the research setting must undergo an in-depth examination of its historical and social background. The results of the case study must describe the social and historical context.
- *Interaction between the researcher and the subjects*: As the researcher and participants interact, data is created. An in-depth reflection is required on the manner in which the data was socially created because the researcher and participants interpret questions and events differently, as every person is unique.
- *Abstraction and generalisation*: The nature of social action and human understanding described by theoretical general concepts must be related to the idiographic detail revealed by the interpretation of data, though applying the *hermeneutic cycle* and *contextualisation* principles. In interpretive research, generalisation is a topic of much discussion. A possible manner in which to handle this is by using Strauss and Corbin's (1990:1) concept of analytical generalisation.
- *Dialogical reasoning*: Some contradictions might arise, which require sensitivity between what the data/findings presented and the theoretical perceptions of the researcher that guided the research design.
- *Suspicion*: The interpretations and narratives (stories) collected from the participants may contain biases, distortions and misrepresentations of what actually happened in the research environment.

The main aim of interpretivists is to prove that their interpretations and observations are "plausible and supported by evidence" (Oates, 2006:295). Criticism of interpretivism includes:

- It discards scientific verification procedures and as a result findings cannot be generalised to other situations (Mack, 2010:8).
- The ontological assumption is that of subjective research rather than objective research (Mack, 2010:8).
- It is not sufficiently radical and it does not acknowledge the ideological and political influences on social reality and knowledge (Mack, 2010:8).

- It is non-scientific – although according to Oates (2006:296) it is intended to be so.
- It is not as well established as positivism (Oates, 2006:296).
- It falls short of analysing the patterns of control and power that regulate the manner in which the world is viewed (Oates, 2006:296).
- Reviewers may not like or know of interpretivism and thus there is the risk that good interpretive research will be rejected for publication because its reviewers were discriminatory against it or had inadequate knowledge to assess it appropriately (Oates, 2006:296).
- In information technology (IT), interpretivism is not widely known (Oates, 2006:296).

In this study, the methodologies of interpretivism and positivism were combined to complement and strengthen their weaknesses through a mixed methods research paradigm. Although critical social theory was not utilised in this study, it will be explained in brief in the section to follow.

4.2.3 Critical social theory

Critical social theory was posited by a German-Jewish philosopher Max Horkheimer (1895–1973) as a social theory oriented towards analysing, evaluating and changing society not partially but as a whole, which is in contrast to other theories that only seek to explain or understand the society. Oates (2006:296) states that critical social research in computing “is concerned with identifying power relationships, conflicts and contradictions, and empowering people to eliminate them as sources of alienation and domination”.

The German philosopher Karl Heinrich Marx (1818–1883) influenced critical social theory. Marx emphasised that capitalism had an oppressive nature caused by economical circumstances. Sahakian (1968:247–248) explains that Marx applied the principle of Hegelian dialectic – consisting of three stages: a thesis (giving rise to its reaction), an antithesis (which contradicts the thesis), and the synthesis (which resolves the tension between the two) – to socio-economic history and identified the bourgeoisie (capitalists – citizens of wealthy urban classes) and proletariat (citizen of lowest/working class) as two conflicting classes. Marx argued that this constant class conflict could only be resolved if the lowest/working class defeated the capitalists to establish a class-free society.

Building on interpretivism, critical social researchers believe that social reality is formed and has objective characteristics according to which the shared understanding of the group is explained and the dominant way of understanding the world is described. Howcroft and Trauth (2004:195) state that critical social research has five characteristics, namely emancipation, critique of tradition, non-performative intent, critique of technological determinism, and reflexivity.

The central possible weakness of critical social research is that it is dependent on change of social realism. According to Stahl (2008:21), “critical research in information systems rarely explicitly incorporates ethical theory. For critical research as ethical research this is not tenable. Critical researchers must render their ethical assumptions and presuppositions open to scrutiny”.

One main difference between the positivistic, interpretivistic and critical social theory paradigms is the researcher’s objectivity. In positivism, the researcher is objective, meaning that he/she does not influence the environment under investigation when data is gathered. In interpretivism, the researcher studies as much as possible about the context and history of the environment under investigation and he/she is allowed to have his/her own interpretations. If the environment under investigation is well understood by the interpretivistic researcher, it will result in valuable and accurate interpretations. The critical social researcher not only has his/her own interpretations of the environment under investigation, he/she also effects change in the environment to determine reaction to this.

A mixed methods paradigm was adopted as the research paradigm in this study in order to address the weaknesses of the paradigms explained above. For this reason, mixed methods research will be explained in more detail than the other paradigms explained.

4.2.4 Mixed methods research paradigm

Mixed methods is the youngest of the paradigms described in this study, as it has only been used in the last fifty years. It is still “in the process of developing its philosophical and theoretical base” (Giggings, 2006:197). The utilisation of mixed methods is growing in popularity and it is viewed as the “third research movement” that is “logical and practical” without entering into the paradigm conflict debate. Multiple research approaches are “answering research questions” instead of restricting the researcher’s paradigm choice (Johnson & Onwuegbuzie, 2004:17).

Mixed research was first introduced by Todd D. Jick (1979:602). Since then, numerous definitions of “mixed methods research” have been offered. Some of the most recent and relevant of these is presented in Table 4.1.

Table 4.1: Definitions of “mixed methods research”

Source	Definition
Johnson & Onwuegbuzie (2004:17)	Mixed methods research can be defined as “the class of research where the researcher mixes or combines quantitative and qualitative research techniques, methods, approaches, concepts or languages into a single study”.
Bazeley (2008:133)	“The term ‘mixed methods’ has developed currency as an umbrella term applying to almost any situation where more than one methodological approach is used in combination with another, usually, but not essentially, involving a combination at least some elements drawn from each of qualitative and quantitative approaches to research”.
Bban (2008:339)	Mixed methods research can be defined as a “combination of at least one qualitative and one quantitative component in a single research design, aiming to include the benefits of each method by combining them”.
Creswell, Fetters & Ivankova (2004:7)	Mixed methods research “involves the collection or analysis of both quantitative and/or qualitative data in a single study in which the data are collected concurrently or sequentially, are given a priority, and involve the integration of the data at one or more stages in the process of research”.
Greene, Caracelli & Graham (1989:256)	Mixed methods designs are “those that include at least one quantitative method (designed to collect numbers) and one qualitative method (designed to collect words)”.
Hewson (2006:179)	Mixed methods research can be defined as “the combination of both quantitative and qualitative methodologies within the same study in order to address a single research question”.
Johnson, Onwuegbuzie & Turner (2007:123)	Mixed methods research “is the type of research in which the researcher or team of researchers combines elements of qualitative and quantitative research approaches (e.g., use of qualitative and quantitative view points, data collection, analysis, inference techniques) for the broad purposes of breadth and depth of understanding and corroboration”.
Leech & Onwuegbuzie (2009:273)	“Mixed methods designs are those that integrate quantitative and qualitative approaches in a single study or a multi-phased study, comprising the following five specific designs: sequential studies, parallel/simultaneous studies, equivalent status designs, dominant – less dominant designs and designs with multilevel use of approaches wherein researchers utilise different techniques at different levels of data integration.”
Rocco, Blis, Gallagher & Perez-Prado (2003:19)	“Mixed methods research combines theoretical and/or technical aspects of quantitative and qualitative research within a particular study.”
Tashakkori & Creswell (2007:4)	Mixed methods research is an enquiry “in which the investigator collects and analyses data, integrates the findings, and draws inferences using both qualitative and quantitative approaches or methods in a single study or programme of inquiry”.

Source: Adapted from Ngulube (2010:254)

Based on the definitions provided above, the aim of mixed methods research is to merge quantitative and qualitative research methods in one study in order to create a research method that is stronger, richer and more reliable by combining the individual research methods strengths and by addressing their individual weaknesses (Johnson & Turner, 2003:299; Mingers, 2001:240; Teddlie & Tashakkori, 2003:696). “[T]here are many ways to mix methods and many levels of mixing both qualitative and quantitative elements in research projects.” (Rocco *et al.*, 2003:20).

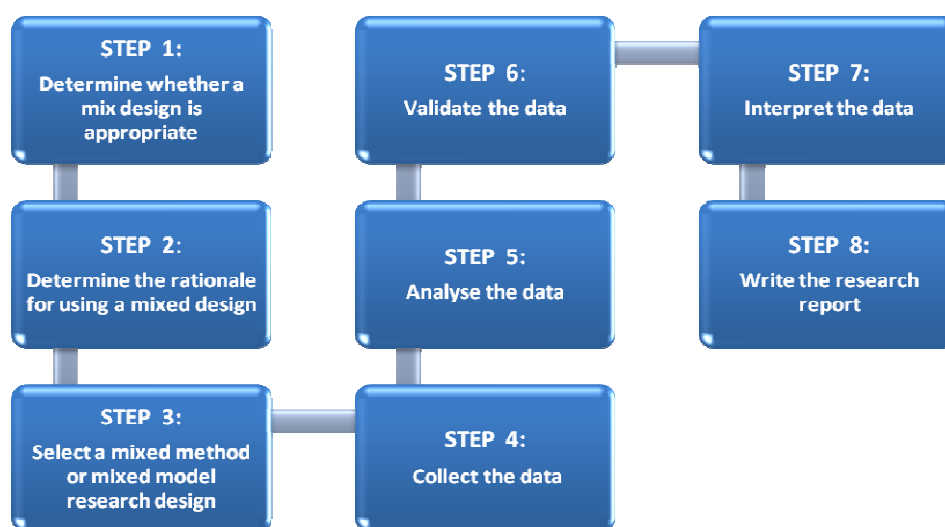
After completing a review of 57 mixed methods research studies, Greene *et al.* (1989:258–259) identified five purposes for applying mixed methods research:

- *Triangulation*: The aim is to converge, corroborate and validate findings or results from different research methods. Triangulation is broadly defined by Denzin (1978:291) as “the combination of methodologies in the study of the same phenomenon”. Triangulation would yield a more valid result than either method could yield independently (Al-Hamdan & Anthony, 2010:51). Different types of triangulation exist, which include data sources-,

investigator-, methodologic-, theoretical-, and data-analysis triangulation (Thurmond, 2001: 254).

- *Complement*: The aim is to strengthen, enhance, elaborate, illustrate and classify the results or findings from one research method with the results or findings from another research method.
- *Development*: The aim is to use the results or findings from one research method to develop the other research method.
- *Initiation*: The aim is to discover contradiction and paradox in one research method using questions or results or findings from the other research method.
- *Expansion*: In order to broaden or extend the scope of the investigation, different research methods must be applied to the different components of the investigation.

There are important steps used in mixed methods research as presented in Figure 4.1. Although the steps are numbered, researchers can follow the steps in the order they require, based on “what particular needs and concerns arise during a particular research study” (Du Plessis & Majam, 2010:468).



Source: Du Plessis and Majam (2010:468)

Figure 4.1: The process stages of mixed methods research

Two types of mixed methods research exist, namely mixed model and mixed methods (Du Plessis & Majam, 2010:465). Mixed model research mixes qualitative and quantitative research approaches across at least two different stages (across-stage mixed model) or within one or more of the different stages (within-stage mixed model) in the research process (Du Plessis & Majam, 2010:465). Mixed methods research emphasises that a separate qualitative and quantitative research approach is included in one study.

Mixed methods research offers both risks and opportunities and it has the ability to provide an inventive opportunity to address research questions (Denzin, 1970:25). Some strengths of mixed methods research include:

- it provides a better understanding of the research problem (Dunning, Williams, Abonyi & Crooks, 2007:147; Molina-Azorin, 2011:8);
- provides comprehensive evidence for studying a research setting than either quantitative or qualitative research individually because it can answer a broader and more complete range of research questions (Johnson & Onwuegbuzie, 2004:21);
- can provide strong evidence for a conclusion by converging and corroborating results and findings (Johnson & Onwuegbuzie, 2004:21);
- combining qualitative and quantitative research methods results in a more comprehensive knowledge base to inform theory and practice because words, pictures and narrative can be used to add meaning to numbers, while numbers can be used to add precision to words, pictures and narrative (Johnson & Onwuegbuzie, 2004:21);
- by utilising it correctly, a researcher can generate and test grounded theory research effectively (Johnson & Onwuegbuzie, 2004:21);
- can use the strengths of one method to address or overcome the weaknesses in another method by using both research approaches in the same study (Johnson & Onwuegbuzie, 2004:21);
- insight that might be overlooked when only a single method is used can be gained (Johnson & Onwuegbuzie, 2004:21);
- researchers can use different data-collection tools available, as they are not restricted to the types of data-collection approaches normally used in qualitative or quantitative research (Johnson & Onwuegbuzie, 2004:21);
- analysis at multiple levels yields a richer understanding than that obtained from analysis at a single level (Johnson & Onwuegbuzie, 2004:21);
- researcher confidence and understanding of the data, results and findings is increased (Dunning, Williams, Abonyi & Crooks, 2007:147);
- the researcher can use any research method available to address a research question (Dunning, Williams, Abonyi & Crooks, 2007:147); and
- scholars in sociology, education, evaluation and health sciences suggest the integration of qualitative and quantitative research methods because of the value, yield and advantages it provides (Molina-Azorin, 2011:8).

Mixed methods research also has some weaknesses and shortcomings which must be understood:

- Mixed methods research is perceived as requiring more labour and financial resources, because it tends to take longer than anticipated (Molina-Azorin, 2011:7; Johnson & Onwuegbuzie, 2004:21).
- Methodological research purists contend that a researcher should conduct research within either a quantitative or a qualitative paradigm (Johnson & Onwuegbuzie, 2004:21).
- Mixed methods research is still in its infancy; therefore, researchers and scholars still need to work out difficulties in analysing quantitative data qualitatively and interpreting findings or results that are in conflict with each other (Johnson & Onwuegbuzie, 2004:21).
- The researcher has to learn to combine multiple research methods appropriately (Johnson & Onwuegbuzie, 2004:21).
- It can be difficult for a researcher to execute both qualitative and quantitative research, particularly if two or more research approaches are done simultaneously. In such cases, a research team might be required (Johnson & Onwuegbuzie, 2004:21).
- Effort and resources are required, as well as advanced skill sets (Molina-Azorin, 2011:7).

Ngulube (2010:252) explored the utilisation of mixed methods research in articles published in library and information science journals from 2004 to 2008 in sub-Saharan Africa. The findings of the study revealed that the utilisation of mixed methods research by library and information science scholars was not popular, which demonstrates that scholars are not aware that mixed methods research is uniquely suited for complex IT investigation.

Feather (1999:23) and Burdett (2000:3) provide evidence that mixed methods research offers useful and enriched findings that are in some cases better than only using either a qualitative or quantitative research method. Molinda-Azor (2011:20) studied the use and added value of mixed methods research in management research and concluded that “mixed method articles have a higher impact in terms of citations than non-mixed methods studies”, which demonstrates that there is a rising interest in the utilisation of mixed methods research. Researchers understand the mixed methods research paradigm to be better than the normal quantitative or qualitative paradigms, as it is viewed as the new research paradigm with the ability to replace older research approaches (Chen, 2006:82). The mixed methods research paradigm was utilised in this study because it merges positivism and interpretivism, quantitative and qualitative research methods, and data-collection and analysis techniques into one research plan (see Figure 4.3) that is more robust, richer and more reliable because the individual strengths of the research paradigms and methods are combined and their individual weaknesses are addressed.

4.3 Research method

In this section, design science as research method will be explained before explaining the research plan. The reason for providing an overview of design science separately to the research plan is that it will be used as a framework into which all other components of the research plan will fit. The research plan will explain the combined research methods, data-collection methods and analysis techniques and their position in the design science process, as well as their application in this study to address the research objectives.

4.3.1 Design science

The design science (also known as design and create) research method is a systematic form of design used to develop IT artefacts or products. March and Smith (1995:251) explain that *constructs*, *models*, *methods/methodology*, and *instantiations* are the different types of artefacts or products that can be developed. *Constructs* are vocabulary, certain perceptions or concepts used within a defined IT environment, such as data flows, while *models* are an amalgamation of constructs that represent a situation and are applied to assist with the development of a solution and the understanding of a problem, such as a data-flow diagram (Oates, 2006:108). *Instantiations* are the realisation of artefacts in the surrounding environment and it ensures that constructs, models, and methodologies are put to operation.

In this study, a *methodology* was constructed by utilising design science as research method within the mixed method paradigm. *Methodologies* use IT to provide guidance on what process stages must be followed and which models must be produced to provide solutions that will resolve problems (Oates, 2006:108). These methodologies include ASDMs, such as DSDM, and PMMs, such as APM. The objective of this study was to construct/develop a hybrid APMM that will have the ability to deliver IT projects in a constantly changing business environment by combining the best practices and strengths of these ASDMs and PMMs (studied in Chapters 2 and 3). Design science is applied by completing the design science cycle as presented in Figure 4.2.

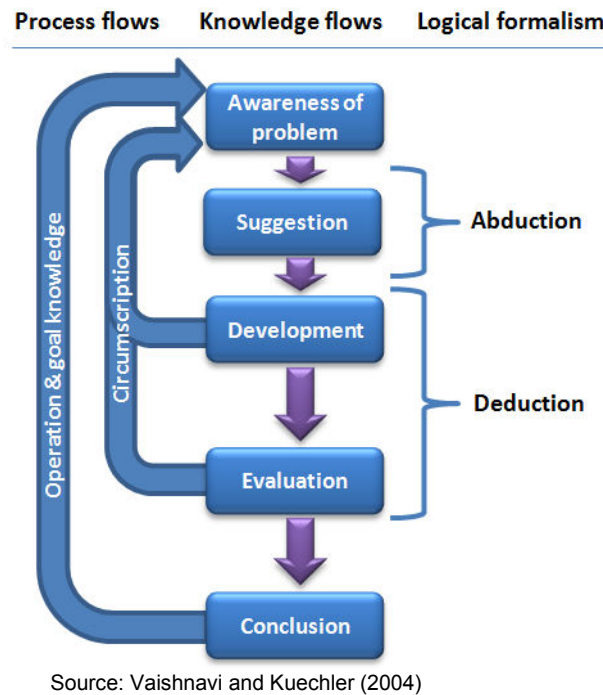


Figure 4.2: Reasoning in the design science cycle

In order to better understand the reasoning of the design science cycle presented above, the five iterative steps are explained. The five fluent steps used in the research plan (see Section 4.3.2) are (Vaishnavi & Kuechler, 2004):

- **Awareness:** The identification and explanation of a problem scenario by studying the literature in which authors recognise areas for further research, or from clients or practitioners expressing the need for something, or from field research, or from new developments and improvements in IT, or studying new findings.
- **Suggestion:** Inventive drive out of curiosity about the problem scenario to offer a very tentative proposal on the manner in which to address the problem.
- **Development:** The tentative proposal is developed and implemented. The manner in which this must be completed depends on the kind of IT product or artefact that must be delivered.
- **Evaluation:** The developed product or artefact is examined and assessed to determine its worth and to determine whether there were any deviations from what was originally expected.
- **Conclusion:** The findings, results and knowledge gained during the design process are documented. Any strange or unexpected results are also identified and may be the subject of possible further research.

In relation to the five steps of Vaishnavi and Kuechler (2004) are the six steps (problem

identification and motivation, definition of the objectives for a solution, design and development, demonstration, evaluation, and communication) of Peffers *et al.* (2007:48) that meets three objectives; it provides a nominal process model, it is consistent with former literature, and it provides a model for evaluating and presenting design science.

Other design science frameworks, guidelines and cycles exist as explained in Nunamaker, Chen and Purdin (1990:99), Hevner *et al.* (2004:75), and Hevner (2007:87).

4.3.2 Research plan

The research plan explains the manner in which the mixed methods research paradigm (interpretivism and positivism), research methods (design science, case study and survey), data-collection techniques (interviews and questionnaires), and data-analysis techniques (cross-case and statistical) were executed in the study. The detailed research plan is presented in Figure 4.3, which demonstrates the manner in which the empirical study was conducted. The design science cycle (see Figure 4.2) was used as the framework into which all other components of the research plan fitted.

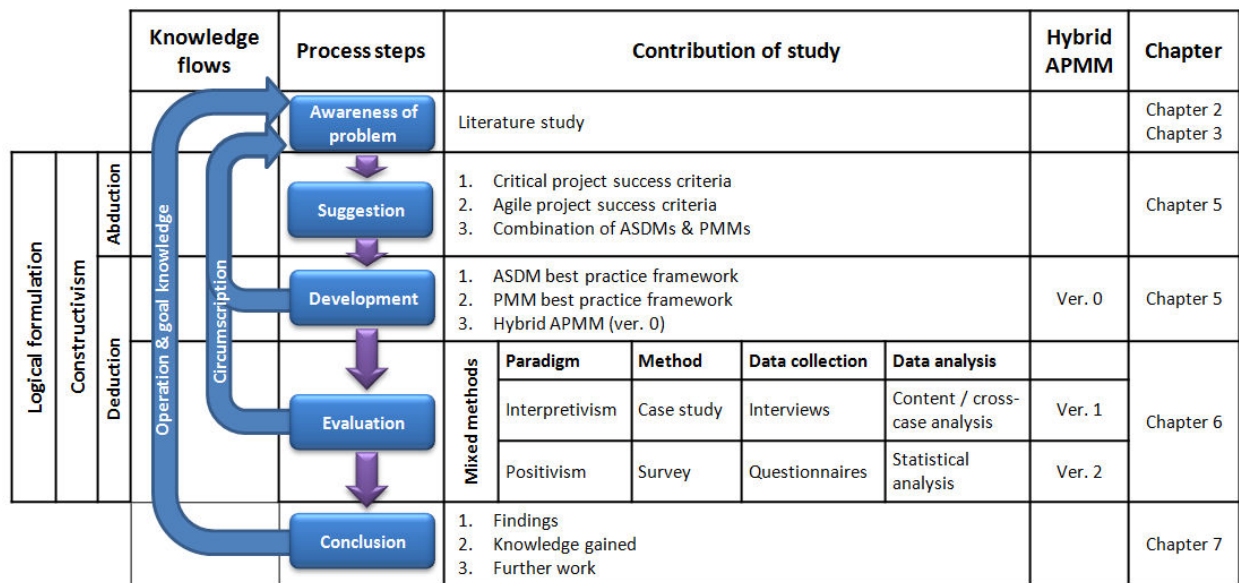


Figure 4.3: The research plan

In order to construct the hybrid APMM, constructivism was applied. Constructivism is an epistemology (theory of knowledge) that claims that people generate their own knowledge and understanding actively by reflecting on their experiences, and not by merely absorbing information from articles and books passively. Constructivists believe that “knowledge and truth are constructed by people and do not exist outside the human mind” (Duffy & Jonassen, 1991:9). This is quite different to objectivists’ claim that “knowledge and truth exist outside the

mind of the individual and are therefore objective” (Runes, 1962:217). Constructivism (see Figure 4.3) was applied in that the hybrid APMM (ver. 0) was constructed by applying the theoretical knowledge gained in Chapters 2 and 3, including the researcher’s experience gained from applying ASDMs and PMMs in practice. The APMM was further enhanced by the knowledge gained after conducting case studies and a survey in organisations of different sizes, which will be reported on in Chapter 6.

As part of the *awareness of problem* process step (see Figure 4.3), a literature review was conducted on the seven ASDMs and three PMMs in Chapters 2 and 3. The possibility of merging ASDMs (studied in Chapter 2) and PMMs (studied in Chapter 3) was investigated by conducting research according to a mixed methods research paradigm, which entailed combining positivistic design science, positivistic surveying and interpretive case study research.

The design science *suggestion* and *development* process steps (see Figure 4.3 and Figure 5.1) were executed in developing the proposed hybrid APMM (ver. 0) that will be presented in Chapter 5. Reasons for the failure of projects in practice and critical project success factors (CPSFs) to increase the probability of a project being completed successfully will also be explained. The CPSFs and reasons for project failure will then be summarised to represent the *critical project success criteria*. The summarised CPSFs were grouped under the APSFs and categories of Chow and Chao (2008:964). New CPSFs identified were combined with the APSFs of Chow and Chao (2008:964) to create the *agile project success criteria*. The different ASDMs and PMMs were evaluated using the APSFs of the *agile project success criteria* to create the *ASDM best practice framework* and *PMM best practice framework*, respectively. The findings for each APSF in the respective frameworks were used as foundation to develop the proposed hybrid APMM (ver. 0) by addressing the weaknesses and combining the different strengths and best practices of each ASDM and PMM. By ensuring that the proposed hybrid APMM (ver. 0) addresses the APSFs, the proposed hybrid APMM (ver. 0) will have a better ability to deliver IT projects successfully.

In the design science *evaluation* process step (see Figure 4.3), the hybrid APMM (ver. 0) was evaluated and improved by conducting an interpretive case studies (see Section 4.3.2.1), which will be reported on in Chapter 6. The case studies entailed interviewing participants in order to create the improved hybrid APMM (ver. 1), after analysing the qualitative data collected using content/cross-case analysis. The hybrid APMM (ver. 1) was also evaluated and improved upon by conducting a survey (see Section 4.3.2.2) through distributing questionnaires to respondents in order to collect quantitative and qualitative data. This data was statistically analysed and the

findings were incorporated into the final proposed hybrid APMM (ver. 2). The findings, knowledge gained and further work will be discussed in Chapter 7.

Data, for example can be generated through interviews, direct observation, documentation and questionnaires. Information was first gathered through a literature review on the different ASDMs and PMMs, which included their effectiveness and utilisation in practice. Qualitative data was collected by conducting interpretive case studies, in which semi-structured interviews were conducted as qualitative data-collection method. Questionnaires were also completed to gather quantitative and qualitative data through a positivistic survey.

The manner in which the design science *evaluation* process step, case studies and survey was conducted will be explained in the sections that follow, after a theoretical explanation has been provided for each.

4.3.2.1 Case studies

A case study examines a natural research setting utilising multiple data-collection techniques to gather data from participants. In a case study, the boundaries of the research setting are not obvious at the beginning of the study and no manipulation or environmental control is done by the researcher (Benbasat, Goldstein & Mead, 1987:370). A case study focuses on one in-depth study of an instance or case, such as an organisation, department or system, of the phenomenon under investigation (Oates, 2006:141). Case studies are characterised by (Oates, 2006:142):

- *Focus on depth rather than breadth:* As much detail as possible is gathered for one instance or case of the research environment under investigation.
- *Natural setting:* The case is analysed and evaluated not in an artificial environment but in its natural environment.
- *Holistic study:* The research aim is not to isolate factors, but to focus on the interrelated, interconnected and complex nature of processes and relationships.
- *Multiple sources of methods:* A wide range of data sources is utilised by the researcher.

Different types of case studies can be applied for the positivistic, interpretivistic, critical social theory and mixed methods research paradigms. There are three basic types of case studies, which can be combined to form six possible case study types (Yin, 2003:5):

- *Exploratory:* This type defines the hypothesis or questions to be used (asked and answered) in the study.
- *Descriptive:* This type results in a detailed analysis of a certain research environment (phenomenon) and its context.

- *Explanatory*: This type “[p]resents data bearing on cause-effect relationships”, describing the manner in which events occurred and the reason for this.

Because case studies focus on the in-depth investigation of one case that represents the research environment, it is important for the researcher to choose and justify the correct case. The selection of a case to be studied depends on (Oates, 2006:144–145):

- *Typical case*: The case can be chosen as representative of the whole class or research environment if it is typical or has the same characteristics of many other cases.
- *Extreme case*: This case is in contrast with and non-typical in relation to the normal cases in the research environment.
- *Test-bed for theory*: This case has characteristics that make it suitable for testing a theory that already exists.
- *Convenience*: This case is convenient when minimum effort, resources and time are required because the participants have agreed to give the researcher access.
- *Unique opportunity*: The researcher was offered the opportunity to study something that was not planned for and that might not occur again.

The researcher has to choose whether he/she will be an active participant through observation or an outside observer. In this study, the researcher was an outside observer in both case studies, which resulted in frank communication, as the researcher was viewed as separate from the companies. The researcher’s own subjectivity is involved as the data is gathered, which is why the researcher must not be viewed as objective when the case studies are conducted (Walsham, 1995:77).

An interpretive research study was undertaken in order to evaluate the proposed hybrid APMM (ver. 0) and determine its applicability, effectiveness and probability of success in practice. The first step to do this was by conducting two case studies in which the proposed hybrid APMM (ver. 0) was distributed to key stakeholders in a large and small organisation with the aim of conducting interviews. The reason for sending the proposed hybrid APMM (ver. 0) to a large (case study A) and a small organisation (case study B) was to evaluate whether it is applicable to different project and business environments. Each organisation formed its own case and their background and setting were different in nature. The cases were both *typical* and *descriptive* case studies.

Case study A was of one of the largest companies in South Africa (and Africa) with more than 35 000 employees that execute projects on the government’s behalf to improve the overall infrastructure and to sustain the economic growth of South Africa. Many industries are directly

dependent on the high-tech projects that are completed by the organisation. This organisation develops and implements some of the largest IT, engineering and construction projects in South Africa, with budgets of R10 billion or more. The Programme Management Office (PMO) was identified as the department to be interviewed and the PMO manager at that time gave written authorisation for a senior programme manager (interviewee A1), project manager (interviewee A2) and team member (interviewee A3) to be interviewed for approximately sixty minutes each.

Case study B was of a small consulting organisation of approximately ten IT consultants, which is responsible for the IT trading platform at one of the major investment banks in South Africa. All the trading done by the bank is directly dependent on the trading system maintained by these consultants. The managing director of the organisation gave permission to interview him as senior manager (interviewee B1), as well as a project manager/senior consultant (interviewee B2) and one of the team members/consultants/programmers (interviewee B3) for approximately sixty minutes each.

Before the interviews were conducted, a hard copy of the questionnaire and hybrid APMM (ver.0) was hand delivered to all the participants and the objective of the study was clearly explained to each participant. This ensured that participants understood the reasons for interviewing them and felt that their opinion was valuable. Each participant was given approximately two weeks to read and understand the proposed hybrid APMM (ver. 0). They also had the opportunity to ask questions by e-mail or telephone if an aspect relating to the hybrid APMM (ver. 0) was not clearly understood. The participants and research setting were not manipulated in any way by the researcher.

Interviews

According to Seaman (1999:562), there are two types of interviews: “a structured interview is described as one in which the questions are in the hands of the interviewer and the response rests with the participant, as opposed to an unstructured interview in which the participant is the source of both questions and answers”. Oates (2006:188) also mentions a semi-structured interview, in which the interviewer still has a list of questions to be asked but he/she is “willing to change the order of questions depending on the flow of the conversation” and can add questions to the list if the participant makes reference to areas that are relevant to the study. Semi-structured and open-ended interviews were conducted by the researcher in interviewing the participants of case study A and case study B in order to collect qualitative data. Semi-structured interviews are normally used in studies where the researcher wishes to discover new findings and it permits the participants to speak freely (Oates, 2006:188).

It is important for the researcher to describe the environment in which the interviews were conducted. According to Walsham (1995:79), the following should be reported on when conducting interviews:

- the characteristics of the research environments selected and the reasons for this selection;
- the number of participants that were interviewed;
- what positions were they employed in;
- what other data sources were utilised;
- the manner in which the data was recorded;
- the manner in which the data was analysed; and
- the manner in which the iterative process between data analysis and theory generation functioned.

The researcher compiled questions beforehand (see Annexure A), which were hand delivered to the participants with a hard copy of the hybrid APMM (ver. 0) and the change request form developed (see Annexure B). The main questions asked in each interview are presented in Annexure A.

As the interviews progressed and participants made reference to areas that were relevant to the study, more questions were added to the list to be asked of the participants.

The interviews were conducted in comfortable meeting rooms at the offices of the participants for both case studies A and B, so that the interviews were not be disrupted. The recording procedure and purpose were explained by the researcher in detail before the interview commenced. If the participants had any questions relevant to the recording procedure or purpose of the interview, these were answered. An MP3 recorder was used with a PC to record the interviews of each participant with a microphone, which was placed on the table between the interviewer and the participant. If the participant wished to hold the microphone in his/her hand, this was permitted. The interview only commenced once the participant felt comfortable and ready.

The manner in which participants perceive the role of the researcher will directly influence the manner in which the participant will respond. For this reason, the researcher must focus on being “professional, polite, punctual, receptive and neutral” (Oates, 2006:188). The researcher realised that some participants were a little intimidated by the recording equipment for the first couple of minutes, although they relaxed as soon as the second or third question was asked. The interviewer tried at all times to show polite interest as the only reaction to questions answered by the participant. Furthermore, the researcher tried at all times not to be judgmental

or to present his own views during the interviews.

All interviews must be transcribed in the same format before content analysis can commence. After the interviews had been completed, they were transcribed into Microsoft Word documents by the researcher in order to be analysed in Atlas.ti 7.0.

Content/cross-case analysis

Content analysis entails studying and analysing text in order to understand the meaning and relations amongst different aspects of the text under investigation. The transcribed interview content was analysed using cross-case analysis as data-analysis technique.

Eisenhardt (1989:541) explains that the main focus of cross-case analysis is to force researchers to examine more than just their initial impressions and it improves the likelihood of reliable and accurate findings. There are three strategies for cross-case analysis (Eisenhardt, 1989:540–541):

- The case studies are divided into two groups based on some attribute or heading and then inspected to see what differences and similarities are revealed between the two groups. The case studies “can be suggested by the research problem or by existing literature, or the researcher can simply choose” some case studies (Eisenhardt, 1989:540).
- Pairs of cases are selected and compared in order to determine and list the differences and similarities identified between each pair. The result of this can be a deeper understanding and identification of concepts not anticipated by the researcher.
- The data is divided based on the manner in which the data was gathered, whether it was by observations, interviews, questionnaires or documentation. This strategy is mainly used when distinguishing between qualitative and quantitative data.

For case studies A and B, three interviews were conducted for each. These interviews (A1, A2, A3, B1, B2 and B3) were analysed by applying cross-case analysis as described by Seaman (1999:567–569).

In case study A, the first transcribed interview (A1) was opened in Atlas.ti 7.0. Interview A1 was reviewed and a list of codes was formulated that described what was being investigated. The codes were aligned with the research objectives of the study to ensure that what was investigated in interview A1 was relevant to the study. Interview A1 was then reviewed and phrases in the text that supported a certain code were listed as evidence under that code. If a phrase contradicted or supported (enhanced) a code, the code was changed to be more explanatory, and the phrase was added as evidence to support the modified code. If a phrase

refuted all codes, a new code was created and the phrase was listed as evidence for the newly created code. All the phrases that supported a certain code were used to form a proposition that represented all the phrases.

The propositions of interview A1 were then used as a starting point for the analysis of interview A2. If a phrase in interview A2 contradicted or supported a proposition in interview A1, the proposition was changed to be more explanatory, and the phrase was added as evidence to support the modified proposition. If a phrase in interview A2 refuted all propositions of interview A1, a new code was created and the phrase was listed as evidence for the newly created code. All the phrases that supported the new code were used to form a proposition that represented all the phrases. This was done for interview A3 as well. The process followed for case study A was also followed for each of the interviews (B1, B2 and B3) of case study B. Case studies A and B thus had their own set of propositions that were supported by their interviews. The cross-case analysis of the different interviews in each case study resulted in a list of propositions and each proposition had a list of supporting and refuting evidence or phrases.

The case studies were then compared using cross-case analysis to determine similarities and differences between the propositions of the case studies. The propositions that supported both case studies A and B, and those that supported either only case study A or only case study B were presented in Section 6.2.2. The findings and recommendations after the completion of the cross-case analysis were used in Section 6.2.3 to enhance the hybrid APMM (ver. 0) before distributing it to respondents with a questionnaire for further evaluation and improvement. The reasons that certain propositions were only addressed by individual interviews were also investigated.

A survey was then conducted in order to enhance the hybrid APMM (ver. 1) and to support the findings and improvements already incorporated into the hybrid APMM (ver. 1) after the case studies had been conducted. The details of this survey will be discussed in the following section.

4.3.2.2 Survey

According to Fink (2003:1), a survey can be defined as “a system for collecting information from or about people to describe, compare, or explain their knowledge, attributes and behaviour”. Using surveys as research method, the researcher is only able to create a shallow but wide perspective in many instances of the phenomenon under investigation (Oates, 2006:141).

The main aim for conducting a survey is to gather the “same kinds of data from a large group of people in a standardised and systematic way” (Oates, 2006:93). There is the perception that only questionnaires can be used to collect data, which is not the case, as interviews, documentation and observation can also be utilised to gather data (Oates, 2006:93).

The planning and execution of surveys can be broken down into the following six activities (Oates, 2006:94–102):

- *Data requirements*: The researcher normally only has one chance to collect data using interviews or questionnaires, which is why it is crucial to consider carefully beforehand what type of data is required, the manner in which it will be analysed and what patterns will be looked for. The data collected should be related to the different areas or topics that address the research question.
- *Data-generation method*: The researcher must make a decision regarding whether interviews, questionnaires or documentation will be used to collect data. For this part of the study, questionnaires were used as qualitative and quantitative data-collection method.
- *Sampling frame*: This is a “list or collection of the whole population of people that could be included in your survey, from which you will choose a sample” (Oates, 2006:95).
- *Sampling technique*: After completing the sampling frame, the manner in which the sample will be selected from the list or collection must be decided upon. There are two kinds of sampling techniques. In *probability* sampling, the chosen respondent sample has a high probability of representing the whole population studied. *Non-probability* sampling provides a fragile foundation for wider population generalisation, as the chosen respondent sample does not represent the whole population. *Probability* sampling techniques include:
 - *Random sampling*: Respondents are randomly selected.
 - *Systematic sampling*: This technique works together with random sampling and entails selecting respondents in regular intervals from the sampling frame.
 - *Stratified sampling*: This technique builds a sample in which the types of respondents are proportional to the whole population under investigation.
 - *Cluster sampling*: This technique entails selecting a group or cluster of people from the population, where the researcher investigates all the members of the selected clusters (Oates, 2006:97). The researcher can cut down on travel costs when focusing his/her study on the cluster or on a sample of the cluster and apply one of the above-mentioned sampling techniques.

Some *non-probability* sampling techniques include:

- *Purposive sampling*: The researcher intentionally hand picks the sample.
- *Snowball sampling*: The researcher selects and collects data from one respondent in the target population, after which the researcher asks for other potential respondents that

are relevant to the research objectives.

- *Self-selection sampling*: The researcher advertises the purpose of the research and the need for respondents. Data is then collected from any respondent.
- *Convenience sampling*: The researcher selects respondents who are easily accessible and willing to participate.
- *Response rate and non-responses*: The response rate to distributed questionnaires is normally low because most people are always just too busy and they view the completion of questionnaires as being of very low importance. For this reason, the researcher must have a strategy in place to ensure a good response rate of about 30%. This can be accomplished, for example, by hand delivering questionnaires and explaining the purpose of the research to the potential respondent.
- *Sample size*: The researcher must make a decision regarding the size of the final sample by estimating the response rate. According to Oates (2006:100), a good rule of thumb is to have a sample set of at least thirty respondents in first-time or small-scale research projects. If there are less than thirty respondents, the statistical analysis will be unreliable and of little value (Oates, 2006:100).

In order to create an improved hybrid APMM (ver. 2), a survey was conducted by distributing the hybrid APMM (ver. 1), including the change request form (see Annexure B), and a questionnaire (see Annexure C) to various practitioners in order to collect quantitative and qualitative data. Purposive sampling was applied as sampling technique and professionals/specialists and project managers of small, medium-sized and large organisations in South Africa who apply IT PMMs or SDMs in some or other form to develop and implement IT projects as part of their daily jobs were selected. The reason for conducting purposive sampling was to ensure that experts in the fields of project management and software engineering completed the questionnaires instead of individuals that do not understand or even use SDMs or PMMs. Purposive sampling ensured that valuable data of high quality was gathered. The selected individuals came from 26 different organisations of different sizes in the following industries, which ensured that a wide variety of specialists contributed towards the survey:

- banking (business and private);
- construction;
- consulting;
- engineering;
- entertainment;
- financial/investment/accounting/auditing;
- IT;
- insurance;

- marketing;
- mining;
- pharmaceutical and medical;
- telecommunication; and
- tertiary education (universities).

Finding respondents who apply PMMs or SDMs in their daily jobs with the relevant skills and practical experience was a time-consuming process. Most time was spent contacting leading organisations across the different industries to ask for assistance or the participation of different managers, IT project managers, IT programme managers, PMO managers, chief information officers (CIOs), chief executive offices (CEOs) and senior software engineers. Many individuals did not wish to provide any assistance, as they were too busy. There were some cases in which managers showed a great deal of interest and provided the researcher with a list of possible individuals to be invited to a presentation session. The researcher had to do a background check to determine whether the potential respondents applied some aspect of SDM or PMM in their daily jobs. The respondents that applied SDMs and PMMs to deliver IT projects were the only ones selected to attend a presentation session.

The hybrid APMM (ver.1) was presented at the selected individual's place of work in a meeting room after the necessary authorisation from management had been obtained. If there was more than one respondent from the same organisation, the researcher tried to schedule a meeting that would suit all the respondents from a particular organisation. Some scheduled meetings were cancelled or postponed because of organisational matters that required attention. Some respondents accepted the meeting but never arrived. After investigating the reason for this, the researcher identified that they were asked to attend more important meetings that were being held at the same time, or they were requested by management to assist in urgent matters. These meetings were rescheduled until these respondents attended a presentation session.

The purpose of the survey, objective of the study and hybrid APMM (ver. 1) were explained to the individual respondent(s) and any questions were answered that were directly related to the purpose of the study. After the hybrid APMM (ver. 1) had been presented, it was e-mailed as a PDF document to the respondents, including the change request form (see Annexure B), with a copy of the questionnaire in Microsoft Excel (see Annexure C).

Questionnaires

Oates (2006:219) defines a "questionnaire" as "a predefined set of questions, assembled in a pre-determined order". According to Brown (2001:6), "questionnaires are any written

instruments that present respondents with a series of questions and statements to which they are to react either by writing out their answers or selecting from existing answers". There are two types of questionnaires (Brown, 2001:6):

- *Self-administered*: These questionnaires are normally mailed out to potential respondents to be completed and sent back when they have the time to do so. This type normally has a low return rate, they must be self-explanatory and clearly understood, and the conditions under which the questionnaire was completed are unknown.
- *Group-administered*: These questionnaires are administered by a group of people, which resolves the issues of conducting self-administered interviews because the respondents will be a captive audience, which creates the feeling that they have to complete the questionnaire. Furthermore, the researcher will be present to answer any questions that there might be and he/she will know the conditions under which the questionnaires were completed.

It is important for a researcher to plan and design a questionnaire correctly before administering it because in many cases the researcher only has one chance to administer the questionnaire to the selected respondents. If the wrong questions are asked or if they are asked out of context, this may lead to responses that are not relevant to the study. The different aspects that have to be considered in questionnaires are summarised below (Oates, 2006:221):

- *Question content and wording*: The researcher must ensure that data relevant to the study is generated. Every question listed in the questionnaire must be relevant, brief, unambiguous, objective and specific.
- *Question types*: Questions can be divided into open questions and closed questions, which normally generates factual data:
 - *Open questions*: The respondent is left to decide what answer to give, by leaving a space under the question.
 - *Closed questions*: The respondents are forced to select an answer from a list or range of answers provided by the researcher.
- *Format of questions and responses*: Questions can be presented in many different ways, for example, yes/no answers, agree/disagree answers, different scale or level of agreement answers.
- *Layout and structure*: The questionnaire must include an explanation about the purpose, manner in which to complete the questionnaire, the return date and return address. It is important for the respondents to feel comfortable in completing the questionnaire by clearly stating that their answers will be kept confidential.
- *Pre-test and pilot*: Before distributing the questionnaire to potential respondents, it can be evaluated by experts in the research field to ensure that maximum value and information

are obtained from the respondents.

- *Validity and reliability:*
 - *Content validity:* The questions asked must adequately address the research objectives.
 - *Construct validity:* The researcher must ensure that he/she is measuring, validating or evaluating what he/she wants to measure, validate or evaluate.
 - *Reliability:* The answers to the questionnaire should be the same if the questionnaire were to be completed again by the respondents.

Group-administered questionnaires were used to gather qualitative and quantitative data. The hybrid APMM (ver. 1) was presented in most cases to a group, although individual presentations also took place. It was sent to the respondents via e-mail in order for them to study the hybrid APMM (ver. 1) before completing the questionnaire. The purpose of the survey and objective of the study were explained again in the e-mail sent out. The respondents were requested to complete and send the questionnaire back via e-mail to the researcher within one week. If the questionnaire was not completed within a week, the researcher followed up via telephone and sometimes even drove though to the respondent's place of work to ensure that as many questionnaires as possible were retrieved. The reason for mainly following up by phone was that most respondents did not reply to any of the reminder e-mails sent out for them to complete the questionnaires. The continuous follow up was rewarded because 78.46% of the 65 questionnaires sent out were returned.

The questionnaire posed five main questions as presented in Section 6.3 and Annexure C. The questions will not be provided in this section but only in Section 6.3 because their composition was dependent on the results of the case studies. For the survey, open and closed questions were asked in the questionnaire.

Statistical analysis

Statistical analysis and techniques assist researchers in analysing data and presenting findings in a universally understandable manner. For example, the average salary per position in a organisation can be viewed by calculating the mean for all people working in that position and plotting it on a graph. The analyst will then be able to see what the highest and lowest paid positions in an organisation are. The data tendency can be determined by applying statistical measures such as mean, median, mode and standard deviation (Oates, 2006:254–258).

The answers to the questions were imported into Excel and STATISTICA in order to conduct a *one-way ANOVA* (Analysis Of VAriance) and a *paired t-test*. The ANOVA will look at the difference in variance of the answers provided for the APSFs by the respondents in the large,

medium-sized and small organisations. The t-test will assess whether the averages of two questions are statistically different from each other. The sorted means for each APSF for large, medium-sized and small organisations will also be analysed and explained. The statistical analysis of the answers to the questionnaires and the results obtained from the analysis will be explained in Section 6.3.1.

The findings from the survey were used to improve the hybrid APMM (ver. 1) to release the final hybrid APMM (ver. 2) that will have a better ability to deliver IT projects in a constantly changing project environment. This will be explained in Section 6.3.2.

4.4 Summary

In this chapter, the positivistic, interpretivistic, critical social theory and mixed methods research paradigms were explained, after which the design science research method was discussed. The manner in which mixed methods research was applied was explained in detail in the research plan by combining positivistic design science, interpretive case studies and positivistic surveying to create versions 0, 1 and 2 of the hybrid APMM.

The development of the proposed hybrid APMM (ver. 0) will be explained in Chapter 5. The *agile project success criteria* will be created and the different ASDMs and PMMs will be evaluated against the different APSFs in the *agile project success criteria* to create the *ASDM best practice framework* and *PMM best practice framework*. The findings and conclusion made regarding each APSF in the respective frameworks were used to develop the proposed hybrid APMM (ver. 0). The practical evaluation and improvement of the hybrid APMM (ver. 0) by conducting a case study and survey will be detailed in Chapter 6.

CHAPTER 5

DEVELOPMENT OF A HYBRID AGILE PROJECT MANAGEMENT METHODOLOGY

5.1 Introduction

In this chapter, the development of the hybrid agile project management methodology (APMM; ver. 0) will be detailed. The manner in which it was developed is presented in Figure 5.1 below.

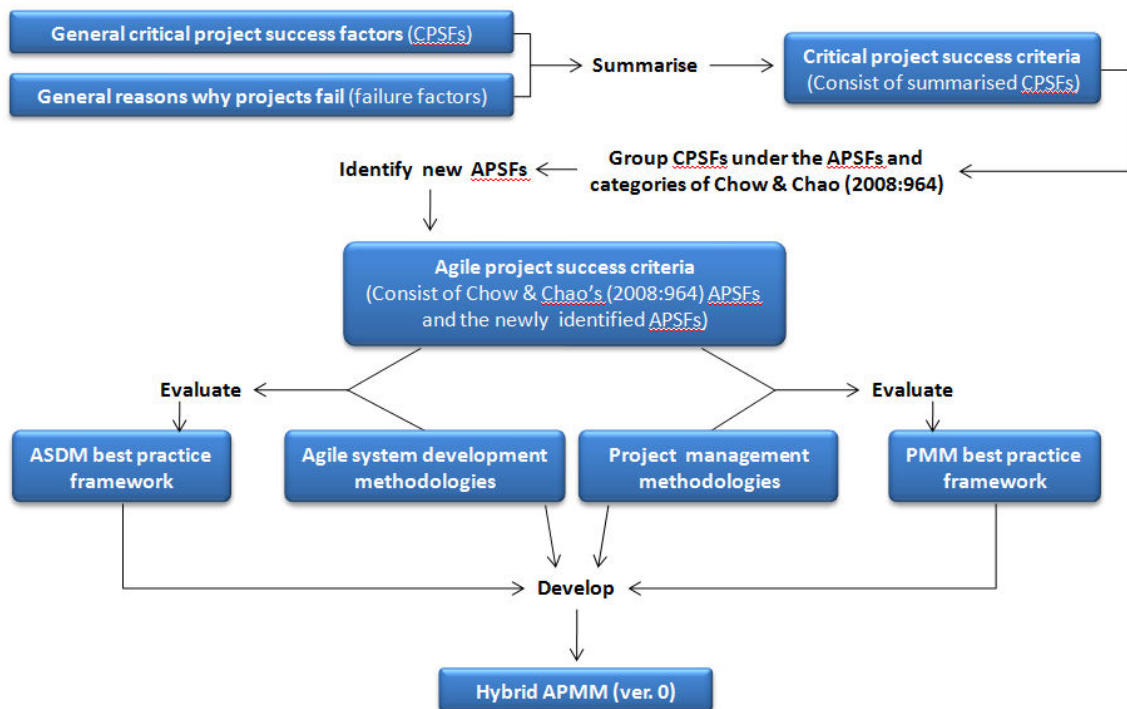


Figure 5.1: The process followed in developing the hybrid APMM (ver. 0)

Theoretical deductions will first be made based on reasons that projects fail in practice and the critical project success factors (CPSFs) required to increase its probability of being completed successfully will be considered. The failure factors and CPSFs will then be summarised to form the *critical project success criteria*. The summarised CPSFs will be grouped under the agile project success factors (APSFs) and categories of Chow and Chao (2008:964). The new CPSFs identified will be combined with the APSFs of Chow and Chao (2008:964) to create the *agile project success criteria*. The different ASDMs and PMMs will be then evaluated using the APSFs of the *agile project success criteria* to develop the ASDM best practice framework and PMM best practice framework, respectively.

Thereafter, the need for a hybrid APMM will be explained. The findings for each APSF were used to develop the proposed hybrid APMM (ver. 0), which will also be explained.

5.2 Theoretical deductions

In practice, many projects fail and many practitioners may claim that their project was completed successfully, even though quality, time, cost or scope constraints were compromised. The manner in which an organisation determines that a project has been completed successfully is not clear. For example, success may be judged by client expectations having been satisfied regardless of the team having overspent to deliver. Success might be defined by the project having been completed according to time constraints but not having satisfied all the client requirements. The perfect project completion scenario would be delivering the project on time within budgetary, quality and scope constraints, and with all customer expectations satisfied.

5.2.1 Why do projects fail?

The very existence of many organisations is reliant on delivering projects successfully in order to generate cash flow and for this reason organisations have an interest in ensuring that projects do not fail. The *reasons that projects fail* and *CPSFs* were investigated to ensure that system development and project management are executed correctly through the hybrid APMM (which addresses these reasons and factors) to complete IT projects successfully. Some of the general reasons that projects fail are relevant to this study:

- unrealistic client and business expectations (Hayes, 2002:196);
- stakeholders have different and unrealistic expectations (Young *et al.*, 2009:12);
- no quality and acceptance criteria (Hedeman, Vis van Heemst & Fredriksz (2010:6);
- poor requirements definition and gathering (Hitech Dimensions, 2002:7; Young, Brady & Nagle, 2009:12; Westland, 2006:xv);
- no clear or unified vision and objectives (Hayes, 2002:196);
- no clear and robust business case or project definition (Hedeman *et al.*, 2010:6; Melton, 2007:9);
- deliverables to be produced are poorly defined (Hedeman *et al.*, 2010:6);
- poorly defined roles, authority and responsibilities (Hedeman *et al.*, 2010:6);
- no robust project delivery plan (Melton, 2007:9);
- no defined checkpoints and structure (Hedeman *et al.*, 2010:6);
- poor testing of product developed (Charvat, 2003:12);
- lack of delivering benefits and value to the organisation (Melton, 2007:9);
- no or poor planning (Hallows, 2005:8; Hayes, 2002:197);
- plans are not used to guide the project or are not executed correctly (Charvat, 2003:12);
- an inappropriate PMM is followed (Charvat, 2003:12);
- no clear methodology is defined and followed (Westland, 2006:xv);
- in practice project management theory is not executed correctly (Charvat, 2003:12);

- inadequate impact analysis is done to determine the impact the project will have on the organisation and business processes (Hitech Dimensions, 2002:7);
- the intended project management strategy is inappropriate (Lock, 2007:17);
- no buy-in or support from the project stakeholders (Westland, 2006:xv);
- no user involvement and commitment from project initiation (Hedeman *et al.*, 2010:6; Young *et al.*, 2009:13);
- no executive ownership (Hedeman *et al.*, 2010:6);
- poor communication (Charvat, 2003:12; Westland, 2006:xv);
- no robust and understandable project scope (Melton, 2007:9; Lock, 2007:17);
- poor project scoping (Hitech Dimensions, 2002:7);
- project scope changes and scope creep (Hallows, 2005:8; Charvat, 2003:12; Young *et al.*, 2009:12);
- insufficient project sizing (Hitech Dimensions, 2002:7);
- incomplete project risk assessment is conducted (Hitech Dimensions, 2002:7);
- projects are approved based on intuitive, personal or political reasons, instead of conducting the necessary risk assessment (Lock, 2007:17);
- there is no real demand for the products that will be delivered by the project (Young *et al.*, 2009:13);
- resource shortages (money, time, expertise, etc.; Hayes, 2002:197);
- inadequate budgets and unrealistic timelines (Thejendra, 2008:125);
- cost and project schedule estimates are too optimistic and they are not revised as the project progresses and new information becomes available (Charvat, 2003:12; Lock, 2007:17);
- no clear understanding of the user and technical requirements and what must be done (Lock, 2007:17; Westland, 2006:xv);
- lack of technology experience (Hayes, 2002:197);
- poor capacity planning (Thejendra, 2008:124);
- project managers are not given training for developing their project management skills which results in the same mistakes being repeated (Charvat, 2003:12);
- poor technical designs and practices (Thejendra, 2008:124);
- organisational politics and culture clashes (Thejendra, 2008:125);
- business decisions made by senior management are commands to be executed, which results in problems and issues being addressed by team members because there is no open communication channel to senior management (Thejendra, 2008:126);
- little support from senior management (Hedeman *et al.*, 2010:6; Hayes, 2002:196);
- poor or inadequate leadership (Young *et al.*, 2009:13);

- weak managers (Thejendra, 2008:125);
- insufficient organisational structure (Thejendra, 2008:126);
- misalignment within the internal organisation (Hitech Dimensions, 2002:7);
- return on investment assessment is not performed (Hitech Dimensions, 2002:7);
- insufficient evaluation of technology (Hitech Dimensions, 2002:7);
- insufficient utilisation of planning tools (Hitech Dimensions, 2002:7);
- insufficient attention is given to the management of cash flow and fund provision (Lock, 2007:17);
- stakeholder concerns and interests are neglected (Lock, 2007:17);
- lack of or inadequate change management (Young *et al.*, 2009:13);
- ineffective change control because of constantly changing business and user requirements (Hedeman *et al.*, 2010:6; Charvat, 2003:12);
- poor control during project delivery (Melton, 2007:9);
- poor quality control (Young *et al.*, 2009:13);
- project problems are identified too late (Young *et al.*, 2009:13);
- too great a reliance on technology (Hallows, 2005:8; Lientz & Rea, 2002:14);
- personal interests are not taken into consideration (Lientz & Rea, 2002:14); and
- external factors, such as vendor and regulatory issues (Thejendra, 2008:126).

In order to develop the best possible hybrid APMM (ver. 0), it is necessary to understand not only the reasons for failure, but also the CPSFs in order to gain an overall view of the requirements for the successful delivery of projects. These will be discussed in the following section.

5.2.2 Critical project success factors

This section discusses factors critical to the success of a project. There is no defined set of CPSFs relevant to IT projects that apply a combination of ASDMs and PMMs, which makes it appropriate to investigate some of the most common and general CPSFs relevant to this study:

- Projects are chosen based on an accurate and sound business plan (Zaval & Wagner, 2009:xxii).
- The project is well defined and there is a sound business case (Lock, 2007:19).
- The benefits, objectives and deliverables of the project are clearly defined and understood (Young, 2007:171).
- A sponsor is assigned, committed and involved in the project (Young, 2007:171).
- Project stakeholders are constantly and continuously informed and consulted on project progress, status and problems (Young, 2007:171).

- Business drivers and project scope are clearly defined and understood (Zaval & Wagner, 2009:xxiii).
- An adequate and skilled project team is put together (Young, 2007:171).
- A project schedule is created and kept up to date (Young, 2007:171; Verzuh, 2008:355).
- Monitoring and control processes and procedures are put in place to track the project and are understood by everyone involved (Young, 2007:171).
- The WBS is maintained and updated (Young, 2007:171).
- The project approval and sign-off processes are controlled (Young, 2007:171).
- There is early risk identification, mitigation, resolution actions and control (Kerzner, 2010:33).
- Project risk is reviewed, monitored and controlled regularly (Young, 2007:171).
- Project issues are resolved promptly and effectively (Young, 2007:171).
- Workable communication and reporting procedures are established (Young, 2007:171).
- There is constant and effective communication (Verzuh, 2008:355; Kerzner, 2010:33).
- Everyone involved benefits from the project (Lientz & Rea, 2002:14).
- A project requires a collaborative effort (Lientz & Rea, 2002:14).
- All the lessons learnt are documented as the project progresses (Lientz & Rea, 2002:15).
- There are sufficient resources and adequate technical competencies (skills, hardware, software, funds, etc.; Lock, 2007:19).
- Resources are allocated proactively (Lientz & Rea, 2002:15).
- There is resource commitment (Kerzner, 2010:33).
- There is careful selection and organisation of the project manager and team members (Lientz & Rea, 2002:15).
- Each team member has a clearly defined role and responsibilities (Kerzner, 2010:33).
- The project depends on its good knowledge base and experienced team members (Kerzner, 2010:33).
- Team members are experienced, skilled and have technical expertise (Kerzner, 2010:33).
- The project team and clients have the same project focus (Verzuh, 2008:355).
- There is support from executive and senior management (Verzuh, 2008:355; Zaval & Wagner, 2009:xxii; Lock, 2007:19).
- Client and users own the project (Zaval & Wagner, 2009:xxii).
- There is client involvement in the early stages of the project (Kerzner, 2010:33).
- Effective change control process is in place and utilised correctly (Zaval & Wagner, 2009:xxiii; Lock, 2007:19).
- The existing business processes are understood (Zaval & Wagner, 2009:xxii).
- A limited amount of experimentation is done with new technologies (Zaval & Wagner,

2009:xxiii).

- The project strategy is well chosen (Lock, 2007:19).
- Effective quality management processes and procedures are in place (Lock, 2007:19).
- Processes and policies are defined with regard to project management best practices and concepts (Kerzner, 2010:33).
- The organisational structure is effective (Lock, 2007:19).
- There is a cross-functional team organisational structure (Kerzner, 2010:33).
- There is cultural awareness and sensitivity within the project (Lientz & Rea, 2002:15).
- The wellness and safety of team members are considered to be of high importance (Lock, 2007:19).
- Team members are motivated (Lock, 2007:19).
- Conflict is resolved quick and fairly (Lock, 2007:19).
- Project schedules are adhered to (Kerzner, 2010:31).
- Quality constraints are adhered to (Kerzner, 2010:31).
- There is a high-quality standard for deliverables and the project as a whole (Kerzner, 2010:33).
- The project budget is adhered to (Kerzner, 2010:31).
- The project schedule is adhered to through adequate planning and frequent tracking of the deliverables (Kerzner, 2010:33).
- Change control processes and procedures are adhered to (Kerzner, 2010:31).
- There are suitable and timeous sign-offs (Kerzner, 2010:31).
- The processes followed are clearly defined and formalised stage gate reviews are executed (Kerzner, 2010:33).
- Scope creep is prevented, and the business and user requirements are well managed (Kerzner, 2010:33; Verzuh, 2008:355).

Kim, Peterson, Chin and Barrier (2001:181–189) investigated possible CPSFs and failure factors to determine whether there are any important differences in terms of the factors' contribution towards the success or failure of information system projects in Korea. The CPSF ranking results of the study is summarised in Table 5.1. The *success rank* column in Table 5.1 represents the ranking of CPSFs required to deliver projects successfully.

Table 5.1: Critical project success factor ranking

CPSFs	Success Rank
1. (Low) User participation in the project	1
2. (Lack of) Clearly stated objectives	2
3. (Lack of) Team-member commitment	3
4. (Lack of) Senior management support	4
5. (Lack of) Project leader's project monitoring	5
6. Project leader's (in)experience	6
7. (Mis)Alignment of project and corporate goals	7
8. Use of (in)appropriate technology	8
9. (Lack of) A detailed project plan	9
10. (No attempt to) Re-engineer business	10
11. (Im)Proper project scope	11
12. (In)Adequate training for team	12
13. Utilising an (In)effective methodology	13
14. (Lack of) Project leader's feedback to team	14
15. (Lack of) Team member's self-control	15
16. Team member (in)experience	16
17. (Little use of) Utilising a prototype	17

Source: Kim *et al.* (2001:186)

In its study, Standish Group International (2001:1) specifies ten CPSFs, which it termed the *CHAOS Ten*, which are presented and prioritised in Table 5.2 below. According to Standish Group International, the more of these factors that are present in a project, the higher the likelihood of delivering a project successfully, although all the factors do not have to be present in order for the project to be completed successfully. The CHOAS Ten were weighted according to their influence on the success of a project. According to this weighting, the higher the CPSF's score, the more important it is to ensure that the CPSF is addressed in a project because doing so would ensure a higher probability of the project being completed successfully.

Table 5.2: CHAOS Ten: Recipe for project success

CPSFs	Score
1. Executive support	18
2. User involvement	16
3. Experienced project manager	14
4. Clear business objectives	12
5. Minimised scope	10
6. Standard software infrastructure	8
7. Firm basic requirements	6
8. Formal methodology	6
9. Reliable estimates	5
10. Other	5

Morris (1988:15–55) and Kerzner (2004:42) both categorised CPSFs into different groups in the same manner as Chow and Chao (2008:964), which will be explained in Section 5.2.4. The reason for not using Morris's (1988:16) or Kerzner's (2004:42) categories to group the summarised CPSFs (see Section 5.2.3) to create the *agile project success criteria* (see Section 5.2.4) is because Chow and Chao's (2008:964) categories are applicable to APSFs specifically.

Morris's (1988:16) and Kerzner's (2004:42) CPSFs were thus kept in this section because they were used together with the other CPSFs explained above to develop the *critical project success criteria* in Section 5.2.3.

Morris's (1988:16) different CPSFs according to each stage of a project are used as a basis in many research studies. These are summarised below:

- *formation stage*: management support, technological advantage, personal ambition, motivated team, clearly defined objectives;
- *build-up stage*: technical skill and expertise, management support, individual and team motivation;
- *main stage*: client and manager support, individual and team motivation; and
- *close-out stage*: individual and team motivation, financial and senior management support.

Kerzner's (2004:42) CPSFs and failure factors in each phase of the project life cycle are shown in Table 5.3.

Table 5.3: CPSFs and critical failure factors according to the project phase

Project phase	CPSFs	Critical failure factors
Executive management acceptance	• Considering recommendations made by employees • Recognising that change is required • Understanding the role of executives in the project	• Not considering associates' ideas or solutions • Not admitting that change may be required • Believing that project management happens at executive level
Line management acceptance	• Placing organisation's interest before that of individual interest • Accepting accountability • Seeking associates advance	• Being unwilling to share information • Not accepting accountability • Being unwilling to seek associates advance
Growth	• Recognising the importance of standardised methodology • Supporting standardised reporting and monitoring • Recognising the importance of good planning	• Feeling intimidated by a standardised methodology, rather than viewing it as a benefit • Not understanding the advantage of project management • Only providing verbal advice during planning
Maturity	• Acknowledging that the project schedule and cost are inseparable • Tracking actual costs • Providing project management training	• Believing that the status of a project is only determined from the schedule • Not viewing cost tracking as important • Believing that project success and growth are the same thing

Source: Kerzner (2004:42)

By addressing the CPSFs and failure factors explained above, the possibility of completing projects successfully can be increased. The factors considered in this section will be summarised to create the *critical project success criteria*.

5.2.3 Critical project success criteria

The reasons that projects fail, CPSFs and failure factors as explained in Sections 5.2.1 and 5.2.2 are summarised as the *critical project success criteria*. The *critical project success criteria* include the following summarised CPSFs:

- *Clearly defined, understood and controlled business case, benefits, objectives, deliverables and project definition:* The business case is used as the main reference document in most projects and it should contain clearly defined project objectives, deliverables and benefits, which define the project.
- *Executive/senior management ownership and support:* The project requires the organisation's support in order to be successful. In order to gain an organisation's support, the project requires the support of executive and senior managers. Management must assume ownership of the project to ensure that it is viewed as an important initiative. A project that is not viewed as important by management is likely to fail or be abandoned altogether.
- *Good quality control, management, processes, procedures, and acceptance criteria:* In order to deliver quality products, the necessary quality management mechanisms must be applied.
- *Adherence to high-quality standards and constraints for deliverables and the project:* In order to deliver a project that is of value to the organisation, the deliverables completed must adhere to the quality standards set.
- *Clearly defined roles, authority and responsibilities:* In order to execute the project, the team members must know what they are responsible for and to whom, and when their tasks are to be completed.
- *Defined checkpoints, gate reviews and structure:* In order to ensure that the project delivers what was originally promised in the business case or project charter, regular reviews and checks on deliverables must be done.
- *Effective change identification, control, management, processes, and procedures:* The project will change on a regular basis and for this reason it is important to have effective change mechanisms in place.
- *Adequate organisational change management:* In order to ensure that there is no resistance to change by the clients/users of the organisation, it is important to have transparent and integrated communication and inform users/employees of project changes that might affect them directly or indirectly.
- *User/client/stakeholder/sponsor involvement, participation, commitment, support, accountability and ownership:* Users, clients, stakeholders and sponsors must be involved in the project in order to be informed of the progress, problems, issues, risks, etc. of the project. By keeping them involved at all times, the probability of delivering a project that

satisfies client expectations is higher.

- *Developing, tracking, adhering to, and correctly executing a detailed/robust project schedule and plan (verbal advice):* By developing, tracking, adhering to and correctly executing a detailed/robust project schedule and plan the project can be guided to successful completion.
- *Sufficient technical design, practices and utilisation of planning tools:* The technical design of a project must satisfy the clients' expectations, as it represents what will be delivered. Planning tools will ensure effective project schedule management.
- *Controlled project delivery:* The release of completed deliverables to the client organisation must be controlled to ensure that there is no/minimised resistance to change from users and employees.
- *Understandable monitoring and controlling processes and procedures to track project progress:* The project must have these processes in place to ensure that the progress is tracked and that it is well controlled and monitored.
- *Stakeholders, users and team members must understand the benefits of successful project delivery for the business:* A project will be viewed as important to the success of an organisation if stakeholders, users and team members understand the benefits a project can bring to the business.
- *Tracking and adhering to cost and budget estimates:* The cost of the project should be tracked regularly to ensure that it does not exceed the approved budget. If changes or circumstances threaten the project's budget, a change request must be logged to ensure that stakeholders are informed and that the necessary decisions are made to ensure that the budget is not exceeded further.
- *Acknowledgement that the project schedule and cost are inseparable:* A project's schedule is directly linked to the cost of the project. If the project schedule is correctly drawn up, every late deliverable means an increase in the project's costs because the estimated cost for completing a deliverable was exceeded.
- *Adequate budgets and realistic timelines:* Before a project is initiated, the budget approved by the client organisation must be adequate to cover all the expenses to deliver the defined deliverables of the project. The timelines for completing deliverables must be realistic, based on the advice of experienced team members and other stakeholders. Similarly, client expectations must be realistic regarding timelines.
- *Trained, experienced and skilled project managers and team members:* The right people with the right skills must be appointed to execute the project. If unskilled or inexperienced people are appointed, the project will be likely to fail.
- *Availability of resources and technical competencies (skills, hardware, software, sufficient funds, etc.):* Information technology projects require technical resources and skilled team

members with the technical expertise to execute a project successfully. If the required resources are not available or cannot be utilised, the project should be suspended until the correct professionals are available to execute the project successfully.

- *Correct application of project management theory:* Project management theory must be correctly applied. Often, practitioners do not execute PMMs or ASDMs correctly when developing or managing a project. If the project then fails, the methodology is blamed, while in reality these methodologies, developed by experts, only work in practice if applied correctly.
- *Correct selection of the correct PMM:* The nature of the project environment must first be analysed to determine which PMM should be used to execute the project successfully. If the wrong PMM is applied, this may result in the failure of a project.
- *Defining and minimising a robust project scope and controlling scope changes:* The scope of the project should be robust and clearly defined. In order to manage change to the project scope, the necessary project control and change procedures must be put in place.
- *Preventing scope creep:* In order to ensure that the scope of a project is well controlled, the necessary control procedures, such as change requests and adaptive actions, should be put in place.
- *Constant and effective communication and feedback:* Communication is the most important aspect of any project and maintains integration of all the components of the project. It is thus important that communication and updates be done regularly, effectively and transparently.
- *Standardised reporting and monitoring procedures:* The manner in which the project's progress, issues, risks, problems, etc. are reported and monitored should be standardised for the whole project.
- *High-quality testing:* Each deliverable must be tested according to a high-quality standard to ensure that a product of value is delivered to the client.
- *Avoiding direction change in the middle of the project:* If the direction of a project is changed in the middle of a project, the impact must immediately be assessed to determine whether the scope, time, cost and quality constraints are still relevant to the project. In most cases, this will not be the case. The business case and project charter must be updated to reflect the new objectives and vision of the project before the project can continue.
- *Managing unrealistic client and business expectations:* Clients might have unrealistic development expectations, while the budget and resources available are not able to satisfy such expectations. In cases like these, a strong project manager is required to explain the concerns to the clients. If the clients still insist upon their requirements regardless of the likelihood of them being satisfied owing to a limited budget and resources, this must be logged as risk on the project.

- *Sufficient resources (money, people, etc.):* Resources are essential to any project. If there are limited resources, the project will suffer or even fail.
- *Proactive resource allocation:* The project manager and stakeholders must anticipate the type of expertise required to undertake certain tasks, as identifying individuals with the suitable expertise might take some time. If new people must be appointed, this may take up to three months, especially if the person needs to resign from his/her job with thirty days' notice. The recruitment process also takes time. If individuals are selected within an organisation to assist with the project, the necessary management approval and time allocation to the project must be done (especially if the person is working on more than one project).
- *Satisfied client, user, business and stakeholder expectations and requirements:* One of the main focuses at all times must be to satisfy client, user, business and stakeholder expectations and requirements within the scope, time, cost and quality constraints of a project. If certain expectations exceed the project's constraints, a change request must be logged, reviewed and approved in order for the necessary changes to be made to the project constraints in order for the new or changing expectations to be incorporated.
- *Clearly defined, gathered and controlled user and business requirements and expectations:* The requirements gathered must be controlled according to the necessary controlling procedures such as change-request forms and adaptive actions to ensure that the project does not lose control. Change-control procedures are not intended for continuous changes to the project owing to team members not assuming their responsibilities. Changes may only be requested if valid proof has been furnished for the urgency of a change to a project. Only clearly defined changes should be executed in order to ensure that there is not much rework and a resulting squandering of time, money and resources.
- *Clear and unified vision/focus:* In order to ensure that the project manager, project team and stakeholders work together on a project to complete it successfully, they must have the same vision or unified focus.
- *Alignment of project and organisational objectives/vision:* The project must be aligned with the strategic objectives of the organisation to ensure that it is of value to its employees and its clients. In cases in which project management and systems development consulting services are provided, it is important to ensure that the project to be completed for the client aligns with the client organisation's strategic objectives and not the consulting organisation's strategic objectives.
- *Consultation with stakeholders/users/clients:* Stakeholders/users/clients are constantly consulted regarding project progress, status, problems and advice.
- *Good leadership:* The project requires a good project manager and organisational managers that can make decisions, give direction and guide the project to successful

completion utilising an SDM or PMM.

- *Effective, aligned and cross-functional team organisational structure:* In many organisations, the team members come from different divisions and have more responsibilities than their responsibilities within the project team. It is thus important that when a project has a cross-functional team that they are kept informed at all times regarding what needs to be completed by whom in order to ensure that the team as a whole is effective.
- *Managing politics and culture:* Managing these factors is difficult in any project. It is important to treat each person involved in the project with respect, especially with regard to his/her political beliefs, culture and unique personality.
- *Managing external project factors, such as vendor and regulatory issues:* Vendors must be effectively managed by the project manager because if they do not deliver on the time, cost and quality expectations provided by the project manager, this could affect the successful completion of deliverables significantly, as well as the project as a whole. Other external factors, such as regulatory issues, must be managed to ensure that the project adheres to the regulations and standards of the organisation and other bodies with which the client organisation is affiliated.
- *Performing a return on investment assessment:* The investment an organisation makes in executing a project must yield a worthwhile return. Before a project is initiated, the return on the investment must be calculated.
- *Sufficient project sizing:* The size of the project must first be determined before a PMM can be applied to ensure that the optimum number and combination of resources can be allocated to the project. If the sizing of the project is inaccurate, this could either result in time and money being wasted on resources that have been over-allocated to a project, or in a project that is understaffed and without the required number of resources (computers, software, hardware, etc.) to execute the project.
- *Regularly and promptly identifying, assessing, mitigating, resolving, reviewing, monitoring and controlling project risks and issues:* These mechanisms must be put in place to ensure that risks and issues are effectively managed.
- *Impact analysis of the project on the organisation:* It is important to assess the impact a project will have on the business processes of an organisation. If business processes will be streamlined, the necessary organisational change management and communication is crucial to the success of the project and growth of the organisation.
- *Appropriate project strategy:* The project strategy must be aligned with the organisation's strategy to ensure that the project is of value to the organisation.
- *Adequate cash flow management and fund provision:* If there are no funds or if the funds are not provided by the project sponsor, there will be no project. The sponsor and project

manager must ensure that the cash flow of the project is adequately managed and that the necessary funds are available when required.

- *Resolving project problems identified as soon as possible:* Once a problem has been identified, it should be resolved as soon as possible. If a serious problem has been identified late, it should be given priority in being solved.
- *No overreliance on technology:* Team members should not depend too greatly on technology to complete their tasks. For example, in some cases team members only use e-mail as a form of communication, even though they see each other in meetings during the day but do not discuss the issues or problems raised in e-mails on these occasions. If the mail server is down for a day or two, or if other people were in meetings the whole day they cannot respond to an issue or problem, which might have been raised in face-to-face communication.
- *Appropriate use and evaluation of, and experimentation with technology:* It is important for an organisation to use the most up-to-date technology to ensure that they remain competitive in an environment in which technology changes regularly.
- *Considering personal interests and ambition, but not placing these before the organisation's interests:* The most important resource of any project is the people associated with the project and organisation, as they are the individuals that undertake and complete the work on the project. If people are appreciated and rewarded for good work, they will work harder and more effectively because they have a sense of purpose and appreciation. However, one individual's interests should not affect the organisation's interest, especially when they are not aligned.
- *Understanding the role of executive/senior management in the project:* The project team and executive/senior management need to understand that executives and senior management are responsible for providing overall guidance to the project to ensure that the project's value is realised and that the project is aligned with the strategic direction of the organisation, as they represent the organisation as whole.
- *Use and acceptance of the benefit of a defined, effective and standardised methodology:* Standardised and internationally accepted methodologies are worth consideration, as they have been developed and improved over many years. If the whole methodology is not applicable to a certain project, it is important to determine whether there are some parts of the methodology that can be applied to the management of a project.
- *Considering stakeholders'/employees' recommendations, ideas and solutions:* By keeping the stakeholders and employees involved in a project the project team can make use of their ideas and possible solutions to project problems.
- *Willingness to share information:* When individuals work in isolation, the progress of a project will be hampered. The team members must be open to sharing experience and

information to ensure that informed decisions are made and that little rework is done.

- *Willingness to seek the advice of senior management:* The project manager must seek direction or possible solutions to problems that might affect the organisation from senior management. For this reason, a firm and confident project manager must be assigned to the project.
- *Understanding the advantages of project management:* The discipline of managing a project ensures that projects are delivered that change people's lives, create jobs, ensure economic growth, etc.
- *Processes and policies are defined with regard to project management best practices, approaches and concepts:* These processes and policies will guide and direct the project in order to ensure that the project adheres to the necessary standards.
- *Knowing that project management does not happen at executive level:* The management of a project is the responsibility of the project manager.
- *Considering the growth and evolution of a project:* Projects evolve because of the changing environment in which they are developed. It is important to continually monitor and control the growth and evolution of the project to ensure that although it evolves, it still is aligned with the organisation's vision, and that it still provides the benefits for which it was originally undertaken.
- *Maintaining and updating the WBS:* The WBS breaks the project down into manageable deliverables and activities. Team members can then be assigned to each activity and deliverable. In order to ensure that all the work to be completed is defined and that every person involved in the project knows what their responsibilities are, the WBS must be maintained and updated if any changes occur.
- *Maintaining the project approval and sign-off processes:* The project manager must ensure that regular sign-offs occur in order to ensure that the client does not later feel that something was not completed according to specification.
- *A project requires a collaborative effort:* The team members must work together to increase the probability of a project being completed within the project constraints.
- *Gathering lessons learnt as the project progresses:* Lessons learned must be logged and communicated as the project progresses to ensure that the same mistakes are not repeated in the project and within other projects of the organisation. If lessons learnt are only logged at the end of the project, the chances of the same costly mistake being repeated in the project and other projects within the organisation will be higher, which might have been prevented if logged as the project progressed.
- *Correct team and project manager selection and organisation:* If a project manager and team who lack the required skills are selected, the project will be a failure.
- *Understanding existing business processes:* If the team members understand the existing

business processes, it will be easier to incorporate the project into and adapt the project to the business environment. Problems and issues in the business processes can also be identified quickly and addressed, corrected and streamlined to make the business more effective in its daily activities.

- *Projects must be selected based on an accurate and sound business plan:* The business plan is used as one of the main reference documents of a project. Most organisations prioritise and select projects based on the information in the business plan. If the business plan is inaccurate, the wrong project might be executed.
- *Clearly defined and understood business drivers and objectives:* In order for the project to align with the strategic direction of an organisation, the business drivers and objectives must be identified, defined and understood.
- *Attempt to re-engineer business:* The business processes must be updated, re-engineered and streamlined to ensure that the business functions effectively and optimally. The re-engineering of processes might cause certain deliverables and even the project as a whole to be completed quickly, efficiently and more easily.
- *Wellness and safety of team members are considered to be of high importance:* If people feel that they are cared for, they will work harder and more effectively.
- *Motivated team members:* Motivated individuals work hard, as they have a sense of purpose. Team members can be kept motivated by having regular social connection sessions and award sessions to reward individuals for work completed according to expectations.
- *Quick and fair resolution of conflict:* Conflict must not be ignored. If conflict arises, a meeting must be scheduled for the parties concerned to voice their grievances. A mediator (project manager) will then help them seek to resolve these quickly and effectively in order to ensure that the work can continue without time and money being wasted on unresolved issues amongst individuals.
- *Suitable and timeous sign-offs:* There should not be so many sign-offs that they become an administrative burden. The deliverables must be grouped together both logically and functionally for sign-offs. If a time-frame must be provided, signing off of deliverables should happen at least once every six weeks.
- *Team-member commitment:* The team members must be committed to completing the project successfully, as it will be rewarding for them when their good performance is rewarded.
- *Standard software infrastructure:* The reporting and communication infrastructure of a project and organisation should be standardised to ensure that all those involved in the project are kept equally informed. For example, if different e-mail software packages (Microsoft Office Outlook, Novell GroupWise, Google's Gmail) are used by various team

members it cause communication problems, such as meeting invites not being added to different software calendars.

- *Reliable estimates*: Estimates of the cost, time and resources required for a project must be reliable. Estimation must be done by people with sufficient experience in order to ensure that the resulting estimates are reliable. The estimates must also include the lessons learnt from previous projects and other organisations that completed the same types of projects (if available).
- *Self-controlled project team*: The team must be able resolve their own problems and find solutions for project-related problems amongst each other. Micro-management should not be done, as this means that the project team is not able to assume responsibility for completing the work assigned to them.
- *Use of prototyping*: Users who will be using the system to be implemented and clients often do not know exactly what they want. A way to manage this is to release pieces or examples (prototypes) of the way in which the system will work to the users. As soon as users see what can be done, they always want more. This results in users being stimulated to think about their expectations and defining their expectations.

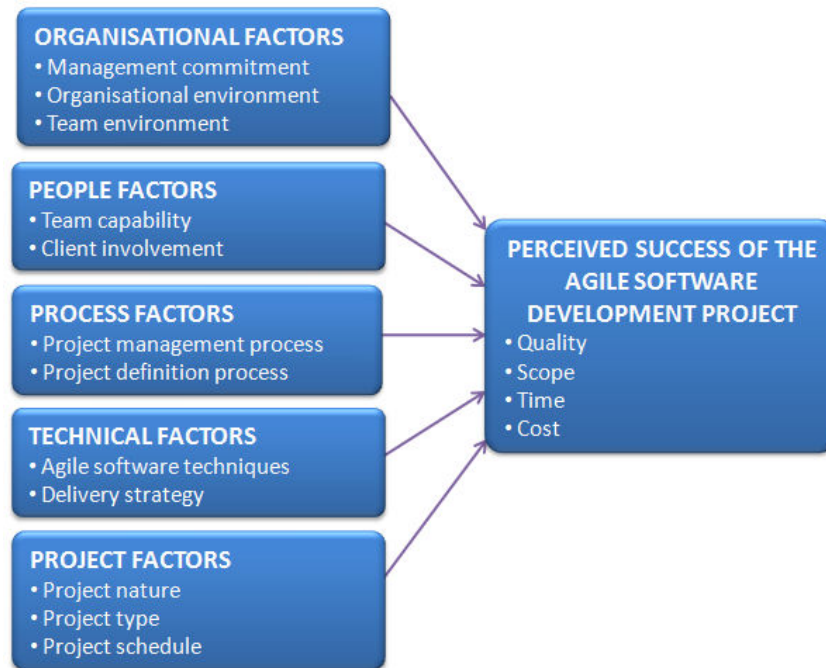
5.2.4 Agile project success criteria

The summarised CPSFs of the *critical project success criteria* explained in Section 5.2.3 were grouped into categories (illustrated in Table 5.4) to determine the area to which these pertain within an agile project environment. As explained in Section 5.2.2, Morris (1988:15–55) and Kerzner (2004:42) classified critical project success/failure factors into different categories based on a typical project's life-cycle stages/phases. This classification was not used in this study. Rather, Chow and Chao's (2008:964) categories were used to develop the *agile project success criteria* because their categories are applicable to APSFs specifically.

Chow and Chao (2008:962–963) conducted research on general reasons for the failure of projects and summarised them as nineteen factors. Regarding the reasons for the success of projects, they determined thirty-six CPSFs for agile projects in particular. Their factors for project failure and agile project success were reduced to thirty-nine factors, which were further reduced to twelve main hypotheses. The twelve hypotheses or APSFs were classified into five categories (organisational, people, process, technical and project) and linked to the four attributes (quality, scope, time and cost) as APSFs required for the successful delivery of agile software development projects (see Figure 5.2).

It is important to note that the reasons that projects fail (Section 5.2.1) and the CPSFs (Section 5.2.2) were obtained from different reference sources than those cited by Chow and Chao

(2008:962–963). The summarised CPSFs in Section 5.2.3 is thus different from the thirty-nine summarised factors of Chow and Chao (2008:963), although there might be some factors that are relevant to each other. The *critical project success criteria* in Section 5.2.3 presented the general reasons that project fails and the general factors of project success – not just factors for agile projects as is the case with Chow and Chao (2008:963). The reason for including general CPSFs are that in essence all projects have some agile elements because many of them have to be delivered in a constantly changing environment.



Source: Chow and Chao (2008:964)

Figure 5.2: The twelve APSFs

Of the twelve APSFs of Chow and Chao (2008:964), only six (delivery strategy, agile software techniques, team capability, project management process, team environment and client involvement) were found by Chow and Chao (2008:969) to be relevant to agile projects, while there was no evidence for the other six APSFs being prerequisites for the success of agile projects. The *agile project success criteria* presented in Table 5.4 show that the twelve APSFs of Chow and Chao (2008:964) are supported by the *critical project success criteria* (Section 5.2.3).

Although only six APSFs were found by Chow and Chao (2008:969) to be relevant to agile projects, the summarised CPSFs were grouped under all twelve APSFs and the five categories of Chow and Chao (2008:964) in Table 5.4 to ensure that all twelve APSFs were addressed in this study. If a general CPSF could not be grouped under the twelve APSFs of Chow and Chao (2008:964), this was added as a new APSF within the corresponding category by highlighting it in red. The *agile project success criteria* were created in this way and are presented in

Table 5.4 on the next pages. Blank cells in the *critical project success criteria* column in Table 5.4 indicate that there was no general CPSF to support a specific APSF of Chow and Chao (2008:964).

Table 5.4: Agile project success criteria

	APSFs (Chow & Chao, 2008:964)	Critical project success criteria
ORGANISATIONAL	Management commitment	Executive/senior management ownership and support
	Organisational environment	Adequate organisational change management
		Stakeholders, users and team members must understand the benefits of successful project delivery for the business
		Alignment of project and organisational objectives/vision
		Impact analysis of the project on the organisation
		Considering personal interests and ambition, but not placing these before the organisation's interests
		Understanding existing business processes
		Clearly defined and understood business drivers and objectives
		Attempt to re-engineer business
	Team environment	Effective, aligned and cross-functional team organisational structure
PEOPLE	NEW: Project selection and prioritisation	Projects must be selected based on an accurate and sound business plan
	Team capability	Trained, experienced and skilled project managers and team members
		Good leadership
	Client involvement	User/client/stakeholder/sponsor involvement, participation, commitment, support, accountability and ownership
		Consultation with stakeholders/users/clients
	NEW: Role definition	Considering stakeholders'/employees' recommendations, ideas and solutions
		Clearly defined roles, authority and responsibilities
	NEW: Communication	Understanding the role of executive/senior management in the project
		Constant and effective communication and feedback
		Willingness to share information
PROCESS		Willingness to seek the advice of senior management
	NEW: Client expectations	Managing unrealistic client and business expectations
		Satisfied client, user, business and stakeholder expectations and requirements
		Clearly defined, gathered and controlled user and business requirements and expectations
	NEW: Politics and culture	Managing politics and culture
	NEW: Collaborative teamwork	A project requires a collaborative effort
		Wellness and safety of team members are considered to be of high importance
		Motivated team members
		Quick and fair resolution of conflict
		Team-member commitment
		Self-controlled project team
	Project management process	Adherence to high-quality standards and constraints for deliverables and the project
		Defined checkpoints, gate reviews and structure
		Correct application of project management theory
		Correct selection of the correct PMM
		Use and acceptance of the benefit of a defined, effective and standardised methodology
		Understanding the advantages of project management
		Processes and policies are defined with regard to project management best practices, approaches and concepts
		Knowing that project management does not happen at executive level
		Preventing scope creep
		Reliable estimates

PROCESS		Adequate cash flow management and fund provision
		Suitable and timeous sign-offs
		Managing external project factors, such as vendor and regulatory issues
	Project definition process	Clearly defined, understood and controlled business case, benefits, objectives, deliverables and project definition
		Clear and unified vision/focus
		Defining and minimising a robust project scope and controlling scope changes
	NEW: Change management and control	Effective change identification, control, management, processes, and procedures
	NEW: Monitoring and control	Good quality control, management, processes, procedures, and acceptance criteria
		Understandable monitoring and controlling processes and procedures to track project progress
		Controlled project delivery
		* Standardised reporting and monitoring procedures
		Resolving project problems identified as soon as possible
		Maintaining and updating the WBS
		Maintaining the project approval and sign-off processes
	NEW: Reporting and tracking	Tracking and adhering to cost and budget estimates
		* Standardised reporting and monitoring procedures
TECHNICAL		Gathering lessons learnt as the project progresses
	NEW: Risk and issue management	Regularly and promptly identifying, assessing, mitigating, resolving, reviewing, monitoring and controlling project risks and issues
	Agile software techniques	
	Delivery strategy	Use of prototyping
	NEW: Technical designs, practices and tools	Sufficient technical design, practices and utilisation of planning tools
		Performing a return on investment assessment
		No overreliance on technology
		Appropriate use and evaluation of, and experimentation with technology
		Standard software infrastructure
	NEW: Testing	High-quality testing
	Project nature	Avoiding direction change in the middle of the project
		Considering the growth and evolution of a project
	Project type	
	Project schedule	Developing, tracking, adhering to, and correctly executing a detailed/robust project schedule and plan (verbal advice)
		Acknowledgement that the project schedule and cost are inseparable
PROJECT		Adequate budgets and realistic timelines
	NEW: Resource management (availability and allocation)	Availability of resources and technical competencies (skills, hardware, software, sufficient funds, etc.)
		Sufficient resources (money, people, etc.)
		Proactive resource allocation
		Correct team and project manager selection and organisation
	NEW: Project size	Sufficient project sizing
	NEW: Project strategy	Appropriate project strategy

The **standardised reporting and monitoring procedures* CPSF was grouped under two newly identified APSFs (monitoring and control, reporting and tracking), as it is relevant to both APSFs. Of the six APSFs of Chow and Chao (2008:969) found by them to be relevant to agile projects, only one APSF, namely *agile software techniques*, was not supported by any of the summarised CPSFs in the table on the previous pages. Of the remaining six APSFs found not to be relevant to agile projects by Chow and Chao (2008:969), the following five APSFs were supported by the summarised CPSFs in the table and may be prerequisites for the success of agile projects:

- management commitment;
- organisational environment;
- project definition process;
- project nature;
- project schedule.

The other APSF, *project type*, which was found not to be relevant to agile projects by Chow and Chao (2008:969), was not supported by any of the summarised CPSFs. This APSF was however kept in the *agile project success criteria*, lest it were found to be relevant to agile projects in Chapter 6.

Fifteen new APSFs were identified, which were included in the *agile project success criteria*. In the sections that follow, the ASDMs and PMMs will be measured against the twenty-seven APSFs of the *agile project success criteria* in order to determine the extent to which the various ASDMs and PMMs support the various APSFs.

5.2.5 Agile system development methodology best practice framework

The ASDM best practice and PMM best practice frameworks were developed interpretively using constructivism. The ASDMs studied in Chapter 2 and PMMs studied in Chapter 3 were evaluated by comparing whether each of them addresses the *agile project success criteria*. The findings for each ASDM and PMM pertaining to a specific APSF are given in Tables 5.5 and 5.7, respectively. The following evaluation responses are given in these tables:

- **(Y)**es: The ASDM/PMM addresses the APSF almost completely (81%-100%).
- **(P)**artially: The ASDM/PMM addresses the APSF partially (31%-80%).
- **(N)**o: The ASDM/PMM does not address the APSF (0%-30%).

It is important to note that the evaluation was done based on the researcher's interpretation, understanding and experience. The evaluation may have been done differently by other

researchers, as each person's interpretation, understanding and perspective are different. The manner in which a certain ASDM is evaluated is based on the researcher's interpretive and in-depth understanding of the area of application, advantages and disadvantages of ASDMs in general and the different ASDMs' components, effectiveness and practical application in IT and other project and system environments of different sizes and natures. The findings for each ASDM pertaining to a specific APSF are given in Table 5.5.

Table 5.5: ASDM best practice framework

	Agile project success criteria	ASDMs							Findings
		DSDM	Scrum	XP	FDD	Crystal ASDMs	ASD	LD	
ORG.	1. Management commitment	P	P	P	P	P	P	P	Weakness. Senior management must treat the project as urgent and important; otherwise, there will be no commitment.
	2. Organisational environment	P	P	P	P	P	P	P	Weakness. The project must be regularly aligned with the organisation's focus, business processes and organisational change-management procedures.
	3. Team environment	P	P	P	P	P	P	P	Weakness. Team members must be correctly allocated to the project across different organisation divisions and the period of their allocation must be communicated to the rest of the organisation.
PEOPLE	4. NEW: Project selection and prioritisation	N	N	N	N	N	N	N	Weakness. Team members must be correctly allocated to the project across different organisation divisions and the period of their allocation must be communicated to the rest of the organisation.
	5. Team capability	N	N	N	N	N	N	N	Weakness. This can be addressed by employing people with the required skills and expertise.
	6. Client involvement	Y	Y	Y	Y	Y	Y	Y	Strength. Keeping the client involved is a fundamental value of all ASDMs.
	7. NEW: Role definition	P	Y	Y	Y	Y	P	P	Strength. Most ASDMs adequately define the role of every person involved.
	8. NEW: Communication	Y	Y	Y	Y	Y	Y	Y	Strength. Continuous and open communication is incorporated in all ASDMs.
	9. NEW: Client expectations	Y	Y	Y	Y	Y	Y	Y	Strength. The identification and satisfaction of client expectations are incorporated in all ASDMs.
	10. NEW: Politics and culture	N	N	P	N	N	N	N	Gap. XP addresses this with its respect value (Holcombe & Holcombe, 2008:25). Political and cultural issues can be managed by studying human behavioural sciences.
	11. NEW: Collaborative teamwork	Y	Y	Y	Y	Y	Y	Y	Strength. Collaborative teamwork is a fundamental requirement in all ASDMs.
	12. Project management process	P	P	P	P	P	P	P	Weakness. This can be addressed through combining ASDMs with the best practices of project management.
	13. Project definition process	Y	Y	Y	Y	P	Y	P	Strength. Clearly defining what must be achieved by the project is incorporated in most ASDMs.
PROCESS	14. NEW: Change management and control	P	Y	Y	P	P	Y	Y	Strength. Effective change management is a major focus of most ASDMs.
	15. NEW: Monitoring and control	Y	Y	Y	P	P	Y	Y	Strength. All ASDMs have either a monitoring or controlling function.
	16. NEW: Reporting and tracking	Y	Y	Y	P	P	Y	Y	Strength. Continual and rapid feedback/tracking is incorporated in most ASDMs.
	17. NEW: Risk and issue management	P	P	N	N	N	P	P	Weakness. Most ASDMs only incorporate risk management partially. This can be addressed through combining ASDMs with other methodologies that address risk and issue management.
	18. Agile software techniques	Y	Y	Y	Y	Y	Y	Y	Strength. All of these ASDMs are agile software techniques in their own right.
TECHNICAL	19. Delivery strategy	Y	Y	Y	Y	P	P	P	Strength. Continuous incremental integration is incorporated in all ASDMs.
	20. NEW: Technical designs, practices and tools	P	P	Y	P	P	P	Y	Strength. Part of most ASDMs, such as features, story cards, pair programming, delayed commitment, and JAD sessions.
	21. NEW: Testing	Y	Y	Y	Y	Y	Y	Y	Strength. Regular and integrated testing is incorporated in all ASDMs.
PROJECT	22. Project nature	P	P	P	P	Y	P	N	Strength. Crystal ASDMs show the potential to adapt to the nature of a project the best.
	23. Project type	P	P	P	P	Y	P	N	Strength. Crystal ASDMs show the potential to adapt to different project types the best.
	24. Project schedule	P	P	P	P	P	P	P	Weakness. This can be addressed through combining ASDMs with the best practices of project management.
	25. NEW: Resource management	P	P	N	N	N	P	N	Weakness. This can be addressed through combining ASDMs with the best practices of project management.
	26. NEW: Project size	P	P	P	P	Y	N	N	Strength. This can be addressed by Crystal ASDMs.
	27. NEW: Project strategy	P	P	P	P	P	P	P	Weakness. This can be addressed through combining ASDMs with the best practices of project management.

The findings presented in the table above demonstrate that most weaknesses lie in the *organisational* and *project* categories. The *project* category weaknesses were mainly addressed by the findings of the PMM best practice framework, while the *organisational* category weaknesses can be addressed by incorporating some general project management practices. The weaknesses identified in Table 5.5 emphasise that there is a need to combine the ASDMs with other PMMs to increase the ability to deliver IT projects in a constantly changing business and project environment.

The evaluation responses (Y, P, N) for each APSF of the different ASDMs were analysed to determine the extent to which the different ASDMs address the *agile project success criteria*. The results of the analysis are presented in Table 5.6.

Table 5.6: Extent to which ASDMs address the agile project success criteria

	DSDM	Scrum	XP	FDD	Crystal ASDMs	ASD	LD
Yes	10	12	13	9	10	10	10
No	3	3	4	5	5	4	8
Partially	14	12	10	13	12	13	9
Total	27	27	27	27	27	27	27
Yes (%)	37.0	44.4	48.1	33.3	37.0	37.0	37.0
No (%)	11.1	11.1	14.8	18.5	18.5	14.8	29.6
Partially (%)	51.9	44.4	37.0	48.1	44.4	48.1	33.3
Total (%)	100.0	100.0	100.0	100.0	100.0	100.0	100.0

The results of the analysis presented in the table above demonstrate that there is no ASDM that fully addresses the *agile project success criteria*. This emphasises the need for ASDMs to be combined with PMMs to develop a hybrid APMM that addresses the *agile project success criteria*.

5.2.6 Project management methodology best practice framework

The PMM best practice framework was developed in the same manner as the ASDM best practice framework. Each of the APSFs was evaluated using the same evaluation responses (Y, N, P) used for the ASDM best practice framework. The results of this evaluation are given in Table 5.7 on the next page.

Table 5.7: PMM best practice framework

	Agile project success criteria	PMMs			Findings
		PMBOK	PRINCE2	APM	
ORG.	1. Management commitment	P	P	P	<i>Weakness.</i> Senior management must treat the project as urgent and important; otherwise, there will be no commitment.
	2. Organisational environment	P	P	P	<i>Weakness.</i> The project must be regularly aligned with the organisation's focus, business processes and organisational change-management procedures.
	3. Team environment	P	P	P	<i>Weakness.</i> Team members must be correctly allocated across different organisation divisions and the period of their allocation must be communicated to the rest of the organisation.
PEOPLE	4. <i>NEW: Project selection and prioritisation</i>	N	N	N	<i>Weakness.</i> Team members must be correctly allocated to the project across different organisation divisions and the period of their allocation must be communicated to the rest of the organisation.
	5. Team capability	N	N	N	<i>Weakness.</i> This can only be addressed by employing people with the required skills and expertise.
	6. Client involvement	P	Y	Y	Strength. PRINCE2 and APM emphasise continual client involvement.
	7. <i>NEW: Role definition</i>	Y	Y	P	Strength. The roles of everyone involved are well defined in PMBOK and PRINCE2.
	8. <i>NEW: Communication</i>	Y	Y	Y	Strength. Continuous and open communication is incorporated in all PMMs.
	9. <i>NEW: Client expectations</i>	Y	Y	Y	Strength. The identification and satisfaction of client expectations are important in all PMMs.
	10. <i>NEW: Politics and culture</i>	P	P	P	Gap. This was addressed through combining with XP. Political and cultural issues can be managed by studying human behavioural sciences.
	11. <i>NEW: Collaborative teamwork</i>	Y	Y	Y	Strength. Collaborative teamwork is a fundamental requirement in all PMMs.
	12. Project management process	Y	Y	Y	Strength. Each PMM has a detailed project management process in place.
	13. Project definition process	Y	Y	Y	Strength. Clearly defining what must be achieved by the project is incorporated in all PMMs.
PROCESS	14. <i>NEW: Change management and control</i>	P	Y	Y	Strength. Effective change management is a major focus of PRINCE2 and APM.
	15. <i>NEW: Monitoring and control</i>	Y	Y	Y	Strength. All PMMs have a monitoring or controlling function.
	16. <i>NEW: Reporting and tracking</i>	Y	Y	Y	Strength. Continual and rapid feedback/tracking is incorporated in all PMMs.
TECHNICAL	17. <i>NEW: Risk and issue management</i>	P	Y	P	Strength. Risk and issue management is fully incorporated by PRINCE2.
	18. Agile software techniques	N	P	Y	Strength. APM is the only PMM that fully incorporates this factor.
	19. Delivery strategy	P	Y	Y	Strength. PRINCE2 and APM support continual incremental integration.
	20. <i>NEW: Technical designs, practices and tools</i>	Y	Y	Y	Strength. All PMM has well-defined practices executable in practice.
PROJECT	21. <i>NEW: Testing</i>	P	Y	Y	Strength. Testing is integrated across all areas of PRINCE2 and APM.
	22. Project nature	P	Y	Y	Strength. PRINCE2 and APM show the potential to adapt to the nature of a project the best.
	23. Project type	P	Y	Y	Strength. PRINCE2 and APM show the potential to adapt to different project types the best.
	24. Project schedule	Y	Y	P	Strength. PMBOK and PRINCE2 clearly define this APSF.
	25. <i>NEW: Resource management</i>	Y	Y	P	Strength. PMBOK and PRINCE2 clearly define this APSF.
	26. <i>NEW: Project size</i>	N	Y	Y	Strength. PRINCE2 and APM show the potential to adapt to different project sizes the best.
	27. <i>NEW: Project strategy</i>	Y	Y	Y	Strength. All PMMs define a strategy to be followed to manage a project.

The findings presented in Table 5.7 demonstrate that the most weaknesses lie in the *organisational* categories, which can be addressed by incorporating some general project management practices.

The evaluation responses (Y, P, N) for each APSF of the different PMMs were analysed to determine the extent to which the different PMMs address the *agile project success criteria*. The results of the analysis are presented in Table 5.8.

Table 5.8: Extent to which PMMs address the agile project success criteria

	PMBOK	PRINCE2	APM
Yes	12	20	17
No	4	2	2
Partially	11	5	8
Total	27	27	27
Yes (%)	44.4	74.1	63.0
No (%)	14.8	7.4	7.4
Partially (%)	40.7	18.5	29.6
Total (%)	100.0	100.0	100.0

The results of the analysis presented in Table 5.8 demonstrate that there is no PMM that fully addresses the *agile project success criteria*. As with Section 5.2.5, this emphasises the need for PMMs to be combined with ASDMs to create a hybrid APMM that addresses the *agile project success criteria*.

In both frameworks, *politics and culture* was identified as a gap. This APSF is very difficult to address because it is impossible to know every person's personal agenda of what he/she wishes to achieve and the manner (manipulation, seduction, etc.) in which they will do it. In order to manage cultural issues, the people involved in a project must understand that a person's reaction to certain circumstances or situations is partly a result of his/her background or the culture in which he/she was raised. The best suggested solution for incorporating this APSF is to treat people with respect and to respect their viewpoints, culture, religion and political understanding.

5.3 Development of a hybrid agile project management methodology

Using design science (research approach) and constructivism (as explained in Chapter 4), the proposed hybrid APMM (ver. 0) was developed by combining the ASDMs, PMMs, and interpretive results of the ASDM best practice framework (see Table 5.5) and PMM best practice framework (see Table 5.7).

5.3.1 Need for a hybrid agile project management methodology

Nothing is perfect – including the APM explained by Highsmith (2010). All ASDMs and PMMs have something to offer, which can be combined in such a way to include the best of each. This will make the overall hybrid APMM stronger than the individual methodologies because its components will be the best components of the different methodologies studied.

There is a critical need within organisations for an adaptable and flexible PMM that has the ability to deliver projects of different sizes and complexities in a constantly changing environment (PM4DEV, 2007:31; Augustine, 2005:213; Cohn & Schwaber, 2003:10–12). The need for a hybrid APMM was explained in Sections 5.2.5 and 5.2.6, as none of the ASDMs or PMMs were found to address the *agile project success criteria* fully. The hybrid APMM (ver. 0) developed in this study will be described in the next section.

5.3.2 Proposed hybrid agile project management methodology

The hybrid APMM (ver. 0) was developed by combining the strengths, addressing the weaknesses and bridging the gaps of the ASDMs and PMMs as presented in Tables 5.5 and 5.7. This was done to ensure that the hybrid APMM (ver. 0) would address the *agile project success criteria* to ensure that it has the ability to deliver IT projects successfully in a constantly changing project and business environment.

In order to show where the *agile project success criteria* are addressed by the hybrid APMM (ver. 0), the number assigned to each APSF (see Tables 5.5 and 5.7) will be placed behind the different sentences that address a specific APSF. These numbers also serve to indicate the way in which the hybrid APMM (ver. 0) combined the strengths, addressed the weaknesses and bridged the gaps of the different ASDMs and PMMs as presented in Tables 5.5 and 5.7. In addition, the components of the different ASDMs and PMMs used to develop the hybrid APMM (ver. 0) will be explained. The hybrid APMM (ver. 0) explanation addressed all the APSFs of the *agile project success criteria*. The evaluation of how effective the *agile project success criteria* is addressed by the hybrid APMM (ver. 0) will be explained in Chapter 6.

The hybrid APMM (ver. 0) life cycle, which consists of seven phases, is presented in Figure 5.3 on the next page. The reason for using phases is that Highsmith's (2010) APM explained in Chapter 3 uses phases and not processes or process groups like PRINCE2 and PMBOK. The new hybrid APMM (ver. 0) is simplified by only providing phases and explains the integration and relationships amongst the phases, instead of using long process flows and activity lists like PRINCE2 and PMBOK.

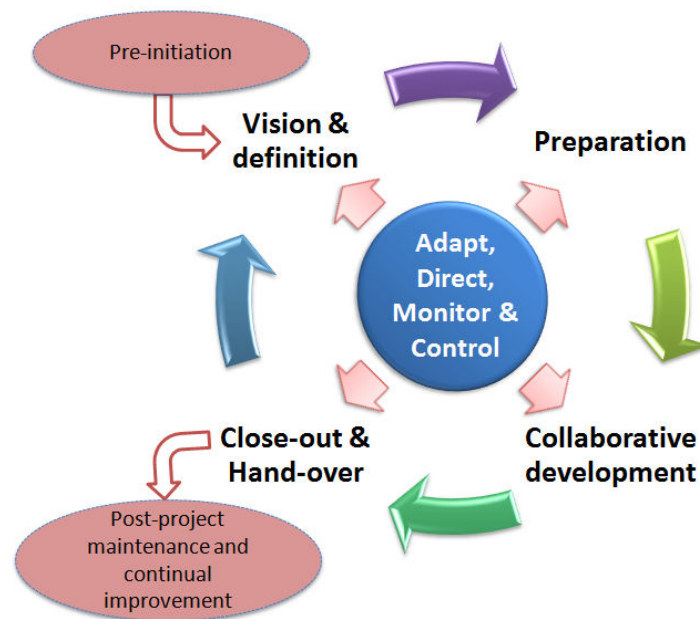


Figure 5.3: The phases of the proposed hybrid APMM life cycle

Each phase will be explained in detail in the sections that follow. The explanations will be with reference to the functions and responsibilities of the different phases.

5.3.2.1 Pre-initiation phase

The *pre-initiation* phase is a combination between the *pre-project* phase of DSDM and the *SU* process of PRINCE2. This phase ensures that everything is in place for the project to be initiated successfully. The key objectives of this phase, some of which were derived from PRINCE2, include the following:

- The initiation of the project is justified in a business case that defines a preliminary project scope, or a preliminary feasibility study is completed to check whether it would be feasible to commence the project (as with DSDM and PRINCE2).
- The necessary senior management and stakeholders are assigned to provide approval on *collaborative increments* completed.
- The different ways in which the project can be developed are evaluated and the approach to developing the project collaboratively is identified.
- People are appointed as part of the project team or as a project manager who will be responsible for executing the work required as stipulated in the project's *vision and definition*. When individuals are appointed from different divisions within an organisation, it is important to ensure that team members are correctly allocated and that the time and duration of their allocation to a project are communicated to rest of the organisation.³ This will ensure that resources are not over-utilised within an organisation.²⁵ The project manager and team members appointed must have the relevant skills and experience to

execute their responsibilities.⁵

- The decision is made regarding whether the project is to commence. Time is not spent on initiating a project that is based on unreliable assumptions and expectations regarding scope, time and cost constraints, and acceptance criteria. The project must only commence if senior management view the project as important and they must give the assurance of and commitment to providing full support and guidance for the duration of the project.¹
- The tasks to be completed during the *vision and definition* phase are *speculated on*.

The project with the highest priority and importance within an organisation should be the project that must be executed next.⁴ This selection and prioritisation of projects is normally completed in the portfolio management function of an organisation, where management ensures that the organisation's strategic goals are achieved by the prioritised projects that must be executed.⁴ This ensures that the projects in execution are aligned with the strategic direction of the organisation.² The business case or feasibility study will then be used as a base-line to develop the project charter in the *vision and definition* phase.

5.3.2.2 Vision and definition phase

This phase is a combination of the *initiation* process of PMBOK, the *IP* process of PRINCE2, and the *envision* phase of APM. Like APM, the project must have a vision that defines what has to be achieved by when, in order to determine whether the project or programme has been delivered successfully (during the *close-out and hand-over* phase). The vision ensures that everyone associated with the project understands the objectives and his/her responsibilities in order for everyone to work towards the same objective as a whole. During this phase, the project vision,¹³ preliminary scope, community^{22, 23, 24, 26} and *collaborative development*^{11, 12} approach (tailored as necessary) are determined. The key objectives of this phase, some of which were derived from PRINCE2 and APM, include the following:

- stipulating the project's vision – what has to be achieved;¹³
- defining the required key capabilities and objectives;⁹
- in order to do benefit realisation in the *collaborative development* phase, the benefits expected, related risks, issues and the reasons for executing the deliverables must be defined;^{13, 17}
- speculating on the time-frame and cost of developing the objectives of the project collaboratively;
- identifying who makes the necessary decisions and provides approval;⁷
- defining the manner in which quality, risks and progress will be measured, controlled and tracked;^{15, 16}
- defining an effective communication structure and levels of authority;^{7, 8}

- tailoring the hybrid APMM (ver. 0) further to suit the project environment;^{22, 23, 26, 27}
- speculating on the activities to be completed during the *preparation* phase;
- empowering the team to ensure that the objectives and capabilities required are delivered successfully;^{3, 11}
- ensuring that the vision of the project is aligned with the vision of the organisation.² (If the project's vision is not aligned with the organisation's vision, it must be re-evaluated and adjusted to ensure that it addresses the organisation's expectations as well.)

In order to adhere to the principle of keeping documentation to a minimum and rather focusing on completing the work, it is necessary to constantly determine the bare essentials of the process and documentation needed (as with APM). There must be a focus on keeping it as simple as possible and not spending time on developing lengthy preliminary project plans and schedules, which are likely to change anyway.

The document that will be completed in this phase is the *project charter*, which will use the business case, or the selected solution in the *feasibility study*, which was completed in the *pre-initiation* phase, together with the key objectives mentioned above, as the base-line. In most cases, the same document used for the *business case* can just be updated after the vision and project's definition have been clarified. The document will justify the project, put major processes of understanding in place, and identify the main stakeholders for the project.^{7, 13} Furthermore, it will define the project's focus, major objectives and deliverables and state by whom they must be delivered within the associated deliverable cost, time and quality constraints.^{7, 9, 12, 19, 27}

5.3.2.3 Preparation phase

This phase is a combination of the *planning* process of PMBOK, the *speculate phase* of APM and the integrated planning processes of PRINCE2. There is a similarity between the hybrid APMM (ver. 0) and APM that a team can speculate based on what is already known, as the future is unknown, although planning cannot be ignored according to Highsmith (2010:129). For this reason, the phase name *preparation* was chosen, as the team must proactively speculate or anticipate on the unknown, while they can plan for what is definite. They can thus proactively prepare for what is known and what is unknown in the *preparation* phase.

Although uncertainty cannot be eliminated through planning – a person can prepare to adapt by anticipating (derived from APM) what could happen should the uncertainty occur, or even before it occurs. For example, when a large project team has to move offices, it can be assumed that it will take the team time to settle into its new environment and that deliverables will be completed

more slowly than usual. The same is true for projects; based on experience, a project manager or team member can predict or anticipate what may happen next based on the current circumstances and project environment. He/she can then proactively prepare beforehand regarding the manner in which to react when something does or does not happen.

The key objectives of this phase include:

- gathering as many requirements as possible beforehand;⁹
- defining the workload by grouping and prioritising deliverables, deliverables sets and incremental release plans;^{9, 19, 25}
- developing an incremental release plan and project schedule guideline;^{19, 24}
- incorporating risk mitigation and issue resolution strategies into the plan;¹⁷ and
- speculating on and estimating project costs, timelines, roles and execution responsibilities.⁷

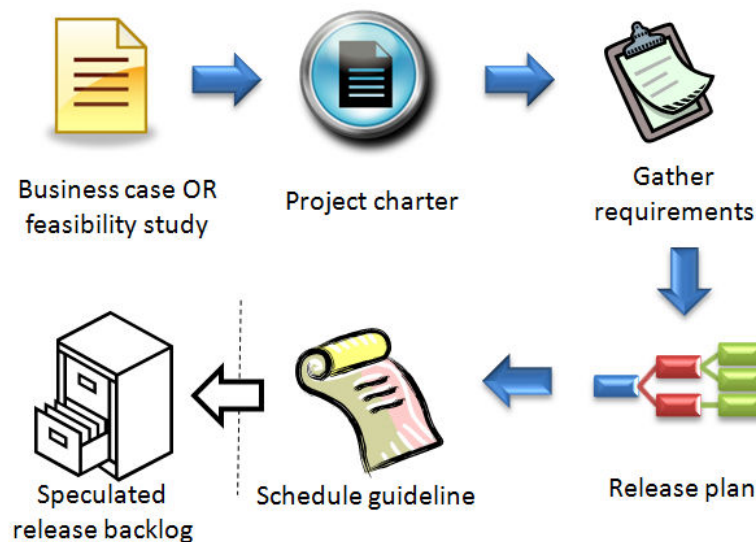


Figure 5.4: The preparation phase

In Figure 5.4 above, the *project charter* that contains the vision and definition¹² of the project is used to gather as many user and business requirements as possible beforehand,⁹ like the feature list of FDD and the product backlog of Scrum. The reason that as many requirements as possible must be gathered before collaborative development takes place is to decrease the possibility of new and changing requirements.¹⁴ Gathering as many requirements as possible beforehand will also ensure that planning and speculation are done based on more known requirements and facts than unknown requirements, which will result in a more accurate and *relevant project schedule guideline*.²⁴ The project schedule guideline consists of assumptions and speculations that will result in the prioritised *speculated release backlog*, which will be the starting point of the *collaborative development* phase.

Requirements are gathered through workshops, JAD sessions (as with ASD), questionnaires or story cards (as with XP).^{6, 8, 9} The requirements are then analysed and explained as *deliverables* and *activities* to be completed in order to satisfy the required client expectations.⁹ The feature breakdown structure of FDD (see Figure 2.9) is then used to categorise, group and prioritise deliverables and activities to create the release plan.¹⁹ In Figure 5.5, the defined *deliverables* and *activities* are grouped into logical *deliverables sets* after the *deliverables* have been prioritised based on their level of importance. The *deliverables sets* are then grouped into *incremental release plans* after they have been prioritised based on their level of importance. The *incremental release plans* are then also prioritised after which the one with the highest priority is selected to be executed as the next *collaborative increment* in the *collaborative development* phase.¹⁹ Each *collaborative increment* is time-boxed, as with ASD and DSDM.

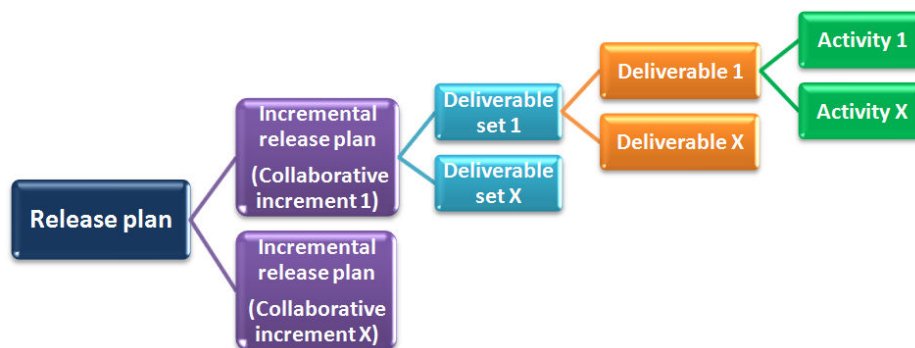


Figure 5.5: The release plan

In order to adapt to the different levels of complexity in projects, one *deliverable* can be completed as part of a *deliverables set*, and one *deliverables set* can even be completed as an *incremental release plan*.^{22, 23, 26} This means that one *deliverable* can be viewed as an *incremental release plan* in small projects.

After grouping and prioritising activities, deliverables, deliverables sets and incremental release plans, the release plan is used to create the *project schedule guideline*.²⁴ The *project schedule guideline* will contain the typical WBS of who does what, how much each deliverable will cost and by when a certain deliverable must be completed.²⁴ The *scheduled guideline* will be used to guide the project as a whole by keeping the *vision* of the project in mind.^{13, 24} Some deliverable dates, deliverables as a whole, and persons responsible for completing deliverables will change as the project progresses. For this reason, it is not called a scheduled plan but a *scheduled guideline*.¹⁴ The *scheduled guideline* will then be used as the speculated release backlog in the *collaborative development* phase. The *release plan* and *scheduled guideline* are built based on the fundamentals of the *functional model integration plan* phase of the DSDM life cycle and the FDD life-cycle processes.^{19, 24}

Customers are involved during every step of the preparation phase to ensure that the team understand the requirements and expectations, and to ensure the client understand the threats and constraints the project team might be facing.⁶ The collaborative phase is directly dependant on the preparation phase.

5.3.2.4 Collaborative development phase

This phase is a combination of the *execution* process of PMBOK, the *MP* and *SB* processes of PRINCE2, and the *explore* phase of APM. Components of ASDMs were mostly combined with the PMMs during this phase because the focus of ASDMs is on the execution and delivery components of a project. The reason that this phase is called *collaborative development* is that the word “collaborative” implies that teamwork, communication and constant stakeholder involvement are very important to ensure the successful development and delivery of each *collaborative increment* and the project as a whole.^{6, 8, 11} *Collaborative development* is also supported by the collaborate activity in the ASD life cycle (see Figure 2.12). Clients are involved and consulted at every step,⁶ and *preparation* (like planning in PRINCE2) is integrated throughout the *collaborative development* phase. The key objectives to be completed during this phase will be explained in the sections that follow.

Iterative refactoring cycle

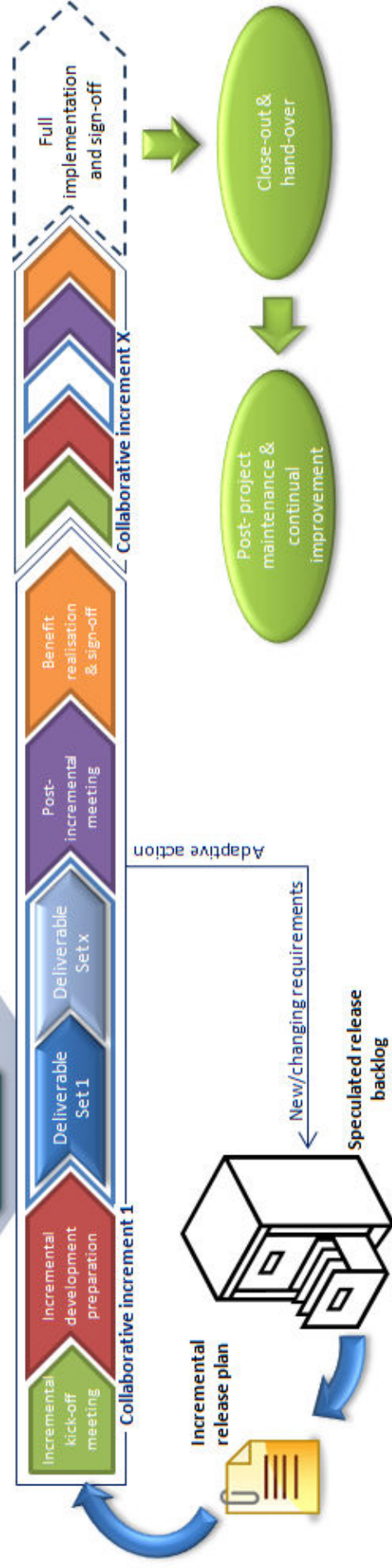


Figure 5.6: The collaborative development phase

The *collaborative development phase* is presented in detail in Figure 5.6. The phase starts with the *speculated release backlog*, which contains prioritised incremental release plans.¹⁹ When an *incremental release plan* with the highest priority is selected, it becomes a *collaborative increment*, which will be executed and signed off to release a new functionality of value that satisfies client objectives and expectations.⁹ The names *speculated release backlog* and *release plan* are derived from Scrum's *product backlog*, the *functional model integration plan* phase of the DSDM life cycle, FDD life-cycle processes and APM.

Every *collaborative increment* is commenced by a kick-off meeting. Everyone involved in executing and approving the *deliverables set(s)* and *collaborative increment* are invited to this meeting in order to ensure that effective *incremental development preparation* (derived from Scrum's pre-sprint planning) is completed by defining the manner in which the deliverables will be executed by whom and by when.^{8, 13, 24} The reason for doing preparation again is that some changes (resources, deliverables, timelines, etc.) might have occurred since the *project schedule guideline* was created in the *preparation* phase,¹⁴ and to ensure that the project's vision is still aligned with the organisation's vision and objectives.² It is important that preparation and speculation happen at a project level during the *preparation* phase and at a *collaborative incremental* level in order to ensure that all new and changed constraints and expectations are taken into consideration before the *deliverables sets* are executed.¹⁴

After completing *incremental development preparation*, every *deliverables set* is executed using the *iterative refactoring cycle* (this name was derived from XP's refactoring practice), in which the design is improved until it is of value.²⁰ The *iterative refactoring cycle* can be run through more than once for a specific *deliverables set*, or for every *deliverable* in the *deliverables set*. More than one deliverable can thus run simultaneously through the *cycle*. The *cycle* commences with a *speculate and analyse* step to ensure that if any changes occurred they are taken into consideration while the project has progressed.¹⁴ The necessary analyses are also completed before the deliverable(s) is physically *created* (or designed).

The deliverable(s) is *evaluated and tested* after the *create* step has been completed.²¹ Here the project manager and project team members determine whether the promises originally made have been fulfilled and thus that the expectations of the users have been met.⁹ If the deliverable(s) does not meet the acceptance and quality criteria, the deliverable(s) is improved during the *improve and adapt* step and tested again during the *test and evaluate* step.^{14, 21} Even if the deliverable passed the *test and evaluate* phase the first time, it is important to check again whether the deliverable cannot be improved or adapted even more within the allowed time and cost constraints to deliver an even better deliverable(s) that would ensure absolute client satisfaction.^{9, 14, 18, 21} In a situation in which a drastic change has to be made to the deliverable(s)

because of *changing or new requirements*, an *adaptive action* (as derived from APM) must be executed to rectify the problem by including the new or changed requirement(s) in the *create* step, which must be executed again with the remainder of the step in the *iterative refactoring cycle*.^{14, 18} The *improve and adapt* phase also ensures that functionalities are included by the project team in a deliverable(s) that clients have not identified and that may result in a failed deliverable(s).^{3, 11}

After the completion of the *improve and adapt* step, the deliverable(s) are reviewed in the *benefit realisation* step (derived from XP and LD) to determine whether the value expected in accordance with the *project schedule guideline* and the client expectations has been realised.^{9, 13, 24} In many projects, deliverables are completed successfully but they do not bring value to the organisation or to the project as a whole. The *benefit realisation* step in the *iterative refactoring cycle* ensures that functionalities of value are released during the *release new functionality* step.¹⁸ The functionalities can be released in the form of prototypes, as with DSDM.¹⁹

Learning is amplified (as with LD and ASD) in the *iterative refactoring cycle* because as deliverables progress through the different cyclical steps many lessons are learnt that must be logged on a regular basis. In the *learn* step, the project team can depend on previous experiences and solutions to problems to ensure that the same mistakes are not repeated during the execution of the next *deliverables set*.^{10, 18} The lessons learnt can also be used during *incremental development preparation* to ensure that future *collaborative increments* do not make the same mistakes as in the past. The lessons learnt can further be utilised at a project level when the *preparation* phase is commenced for a new project in order to ensure that the mistakes made in previous projects are prevented in future projects.¹²

During the *iterative refactoring cycle*, short *daily stand-up meetings* are conducted, which almost have the same function of the daily meetings of Scrum.¹⁸ The reason for having stand-up meetings is to ensure that people do not talk and waste time by discussing matters that are not urgent or relevant to the project when they sit down.⁸ During this meeting, problems are discussed, quick solutions are found, and assistance is provided by the project team if necessary.⁸ Furthermore, it ensures that the team members stay focused on the vision and the task at hand.^{11, 13} Issues and risks are quickly resolved to ensure that any negative impact on the project is minimised.¹⁷

The *direct, monitor and control* process of the *iterative refactoring cycle* is integrated with the *adapt, direct, change and control* phase.^{12, 14, 15, 16, 18} The only difference is that this process occurs at a deeper iterative level of the project and not at an overall project level. This process directs, monitors and controls the *iterative refactoring cycle* as deliverables progress through the

steps of the cycle.^{12, 15} Progress, issues, risks, problems and new functionalities are tracked and reported upon to all stakeholders.^{16, 17} Delayed commitment, as with LD, is applied within the *iterative refactoring cycle* to postpone irreversible decisions for as long as possible without affecting the cost, time and quality constraints until informed decisions can be made based on definite facts instead of predictions.^{18, 20}

After the completion of the *deliverables sets*, a *post-incremental meeting* (derived from Scrum's post-sprint meeting) is conducted, in which the project team, clients and other stakeholders go through a final *benefit realisation* exercise after all the deliverables sets have been completed within a specific *collaborative increment*.^{6, 9} If the *deliverables sets* consist of only one deliverable each, it will not be necessary to go through another *benefit realisation* exercise, as this would have been completed during the *iterative refactoring cycle*.^{22, 23, 26} The *collaborative increment* is approved and signed-off when all the client expectations have been satisfied, benefits have been proven, and value has been realised, as promised in the *project schedule guideline* and the *incremental development preparation* step.^{9, 19, 24, 27} Collaborative incremental delivery (derived from PRINCE2's MP process) is done by having regular sign-offs.^{19, 27} During the *benefit realisation and sign-off*, it is important to check that what has been delivered is still aligned with the organisation's vision.²

During *benefit realisation and sign-off* at a *collaborative increment* level, *new or changing requirements* and expectations can arise that must be added to the *speculated release backlog*.¹⁴ If a significant or large change occurred during the *approve and adapt* step that could not be solved during the *iterative refactoring cycle*, this may result in a *new or changing requirement*, which must be added by an *adaptive action* to the *speculated release backlog*.^{14, 24} After any new or changing requirements have been added to the *speculated release backlog*, these must quickly be regrouped and prioritised before the next *incremental release plan* is selected to be executed.²⁴

It is important to realise that more than one *collaborative increment* can be in execution as the next *incremental release plan* can be selected once all dependencies on previously selected *collaborate increments* have been satisfied. The collaborative incremental boundaries are managed and controlled throughout the life cycle of the project, just like SB and CS processes of PRINCE2.^{12, 15, 18}

After the successful completion and sign-off of each *collaborative increment*, a *full (system) implementation step* can be executed (only if necessary) to integrate the different *collaborative increments* in order to release them as a whole into the organisation.^{19, 27} This step is derived from the implementation phase of the DSDM life cycle. The system and processes are then

handed over to the organisation in the *close-out and hand-over* phase, after which the system must be maintained and continually improved in the *post-project maintenance and continual improvement* phase.¹²

5.3.2.5 Close-out and hand-over phase

This phase is a combination of the *closing* process of PMBOK, the *CP* process of PRINCE2, and the *close* phase of APM.¹² Like all the PMMs, the hybrid APMM (ver. 0) also has a clear end. The *close-out and hand-over* phase is just as important as the other phases of the project. It is not an isolated phase, as it can be applied during any phase of the hybrid APMM (ver. 0). It can also be used to close a project down prematurely for whatever reason (organisational bankruptcy, for example). During this phase, the project is formally closed off (even if the completion of the current project results in the kick-off of another project) by providing the client with all the signed-off deliverables, deliverables sets, collaborative increments and final implementation sign-off documentation (if relevant).^{19, 27} The reason for obtaining formal closure is to ensure that the client or stakeholders do not later claim that certain expectations or functionalities were not provided. Furthermore, the final outcomes of the project must be measured against the vision, value and benefits that were committed to in the *project schedule guideline* and project charter.^{13, 24}

The formal hand-over and training of the project or system is of high importance. In order to minimise documentation, full training can be provided to the users on utilising the system effectively together with a quick reference guide.¹⁸ The guide will refer to different important functionalities that are utilised by the user in his/her respective area of work. The development of a detailed manual is avoided because users normally become lost in voluminous manuals with too much information. An alternative is to create a help file using an index tool that would assist users in finding assistance quickly within their respective area of work. Only when users can utilise the system effectively without any assistance or training can project hand-over be viewed as being complete.

Many organisations require the documentation of detailed design specifications, which cannot be ignored. In a case such as this, it would be wise to consult with the client regarding the level of detail required in order to minimise the documentation workload.

5.3.2.6 Adapt, direct, monitor and control phase

This phase is a combination of the *monitor and control* process of PMBOK, the *DP* and *CS* processes of PRINCE2, and the *adapt* phase of APM.^{12, 15} This phase serves an integrating function, which takes place across all the different phases of the project.

The different phases of the proposed hybrid APMM (ver. 0) have the ability to adapt to their environment.^{22, 23, 26, 27} Projects of different sizes can be executed by replacing the collaborative development phase's *deliverables sets* with only a single deliverable as explained.²⁶ The hybrid APMM (ver. 0) can easily and effectively adjust to changing business and user requirements at a *collaborative incremental* level and iteration (deliverables set or deliverable) level during the *iterative refactoring cycle* by using adaptive actions (as with APM) instead of corrective actions.¹⁴

The stakeholders, management, sponsor or project board (if appointed) supervises by putting certain project controls in place in order to make informed decisions based on facts not assumptions. They have the authority to provide *ad hoc* guidance as the project progresses.¹⁵ A culture of “respect for all” must be fostered together with the project manager, who must continually monitor, control and address any political or cultural issues that might arise as the project progresses.¹⁰ Human behaviour could also be studied, if there is sufficient time, to assist with the management of political issues and cultural behaviours.¹⁰

Like in PRINCE2, the stakeholders, management, sponsor or project board (if appointed) can assign project or organisational assurance to execute some of the assessing and reviewing functionality. They direct the project by considering exceptions as pointed out by the project manager, who acts as a communication channel by communicating all relevant project and progress information to programme or organisational management.⁸

Continual monitoring, measuring and controlling of project-related activities are completed in order to ensure that the project team satisfies the deliverables of the project.^{9, 15} The team continually monitors and measures the progress of the project against the *project schedule guideline* in order to take the necessary adaptive actions if the project's scope, time, cost and quality constraints are threatened.^{14, 15, 16, 24} The key objectives of this phase are to ensure that:

- authority has been provided to initiate the project;
- authority has been provided to sign-off the completed collaborative increments;²⁷
- the project remains viable, and that direction and control are provided throughout the duration of the project (like in PRINCE2);^{13, 15}
- programme and organisational management have an interface to the project;^{2, 8}
- authority has been provided to formally close-out and hand-over the project;^{12, 27}
- deliverables and functionalities are managed and reviewed to realise the post-project maintenance and continual improvement activities to be completed;^{12, 27}
- the project's vision remains focused – what must be delivered by whom and by when¹³ –

any deviation from the original value and benefit agreed upon in the project charter are monitored to avoid divergence from the focus, which might cause scope creep;^{15, 24}

- the problems, cost, quality, issues and risks of the project are controlled;^{14, 15, 16, 17}
- the *project schedule guideline* and structured backlog are reviewed regularly;^{15, 24}
- the defined and accepted collaborative increments are delivered within the quality, time and cost constraints and tolerances to achieve the defined business benefits and value as stipulated in the project charter and project schedule guideline;^{15, 16, 24}
- the project work is monitored and controlled on a daily basis;¹⁵
- the project follows an integrated change control process;
- the project scope, schedule guideline, quality and cost are controlled, monitored and verified regularly;¹⁴
- the performance and progress of the project are tracked and reported on regularly;¹⁶
- procurement is administrated; and
- ensured that all issues are resolved quickly to minimise any negative impact on the project.^{12, 17}

In order to control and monitor a project effectively, it is important to report on the progress of a project in order to identify possible problems and problems that arose, and to identify adaptive actions before they become a threat to the successful delivery of a project.^{14, 15, 16, 17} The escalation process of informing the stakeholders, management, sponsor or project board of identified risks, issues and changes to the project must also be formalised during this process.^{6, 8}

5.3.2.7 Post-project maintenance and continual improvement phase

This phase is partially derived from the *post-project* phase of DSDM. After the implementation, *close-out and hand-over* of a project, maintenance of the implemented solution should continue to ensure that the solution continues to bring value to the client organisation.^{12, 18, 27} In some cases, maintenance is done by the client organisation itself after hand-over.

There is always room for improvement to a solution, which is the reason that it is important to monitor continually whether improvement or an upgrade to the current implemented solution is necessary.¹⁵ As soon as users see what is possible, they always want more. For this reason, there are new releases of software and hardware products in the market to provide quicker and better solutions for clients. The proposed hybrid APM (ver. 0) life cycle is then re-initiated by the *vision and definition* phase if an improvement or upgrade is required. If pre-initiation planning is required, it can be done before the *vision and definition* phase is executed.

5.4 Summary

In this chapter, the possibility of merging ASDMs and PMMs was investigated and the development of a hybrid APMM (ver. 0) was detailed. The reasons that projects fail and the CPSFs were studied and summarised to develop the *critical project success criteria*, which were categorised under the five categories of Chow and Chao (2008:964) to develop the *agile project success criteria* (see Figure 5.1). The *ASDM* and *PMM best practice frameworks* were created by identifying whether a certain ASDM or PMM would satisfy a specific APSF. The findings regarding each APSF in the respective frameworks were used as foundation to develop the proposed hybrid APMM (ver. 0) by combining ASDMs and PMMs in such a way that it would address the *agile project success criteria*.

The hybrid APMM (ver. 0) addressed all the APSFs of the *agile project success criteria* in Table 5.4. It is important to note that there may be other APSFs or *agile project success criteria* used in practice that are undisclosed and not available in the literature, or there may be other APSFs or *agile project success criteria* in literature that may have been overlooked to be included in this study. Whether the hybrid APMM (ver. 0) has addressed the *agile project success criteria* adequately and whether the hybrid APMM (ver. 0) would be of any value in practice will be discussed in Chapter 6. The first step taken in this regard was to evaluate the effectiveness of the hybrid APMM (ver. 0), which will be reported on in the following chapter.

CHAPTER 6

EVALUATION OF THE HYBRID AGILE PROJECT MANAGEMENT METHODOLOGY

6.1 Introduction

The purpose of this chapter is to report on the evaluation of the hybrid APMM (ver. 0) developed in Chapter 5 by conducting case studies and a survey. Case studies were conducted at a large (case study A) and small (case study B) organisation. The content of each interview (A1, A2, A3, B1, B2, B3) was analysed in Atlas.ti 7.0, after which case studies A and B were compared using cross-case analysis as analysis method.

The hybrid APMM (ver. 0) was improved by incorporating the findings of the cross-case analysis. The hybrid APMM (ver. 1) was then distributed to the different respondents across different organisations along with a questionnaire in order to collect quantitative and qualitative data from the respondents. The questionnaires were then statistically analysed and the findings were incorporated into the hybrid APMM (ver. 1) to develop the final hybrid APMM (ver. 2). The evaluation of the hybrid APMM (ver. 0 and 1) will be discussed in this chapter.

6.2 Evaluation of the hybrid agile project management methodology (ver. 0)

The evaluation of the hybrid APMM (ver. 0) was done as explained in Chapter 4. Semi-structured and open-ended interviews were conducted with three participants in case study A and three participants in case study B in order to gather qualitative data. After the interviews had been completed, they were transcribed using Microsoft Word and content-analysis was done in Atlas.ti 7.0. The transcribed content was further analysed using cross-case analysis as data-analysis technique to determine certain correlations and differences in the qualitative data.

6.2.1 Content analysis

The qualitative content of each transcribed interview were analysed in Atlas.ti 7.0. The results of each case study (A and B) will be presented by combining the propositions of the interviews related to a specific case study. The interviews (A1, A2, A3, B1, B2 and B3) that supported a specific proposition will be placed in brackets behind the proposition. The propositions of every case study will be grouped under six themes:

- **Theme 1: Critical project success factors:** General factors that should be addressed and incorporated into a PMM or ASDM to increase the ability of a PMM or ASDM to execute

and deliver a project successfully.

- **Theme 2: *Environmental characteristics*:** These are factors and characteristics that influence the environment in which a project must be delivered.
- **Theme 3: *People factors*:** The interactions between humans that has an impact on the execution and delivery of a project.
- **Theme 4: *Project management and system development principles*:** The principles that should be incorporated to increase the ability of a PMM or ASDM to execute and deliver a project successfully.
- **Theme 5: *Advantages of the hybrid APMM (ver. 0)*:** The identified strengths or advantages of the hybrid APMM (ver. 0).
- **Theme 6: *Shortcomings of the hybrid APMM (ver. 0)*:** The identified disadvantages or shortcomings of the hybrid APMM (ver. 0).

6.2.1.1 Results of case study A (large organisation)

Case study A was executed as explained in Chapter 4. The results of the analysis of the content of interviews A1, A2 and A3 are presented under the six themes below.

Theme 1: Critical project success factors

- *Consider the nature of a project before applying a PMM/ASDM (A2, A3):*
Each project is unique, which is the reason that the project's nature and environment must first be analysed and understood before applying a specific PMM/ASDM. Interviewee A2 stated that "you need to consider the nature of the project before you decide on a specific methodology", while interviewee A3 emphasised that "it's all about scenario analysis, it sounds like a long process but it isn't really".
- *Deliver quickly (A1):*
In practice, there is pressure on project managers and team member to deliver quality products as quickly as possible. Interviewee A1 explained that a project team requires an environment in which they can be creative and "think outside the box", an environment in which they can "effectively and efficiently deliver quickly".
- *Do benefit realisation regularly (A1, A2, A3):*
All three interviewees were in agreement that it is important for the project manager to check regularly that the project team delivers what was promised when the project started. Interviewee A1 stated: "I will need to deliver on the promises that I've made". She furthermore explained: "how can you not do benefit realisation and then at the end of the day not even deliver after receiving all the money?". Interviewee A2 explained that benefit realisation "is very important because right when you start off with the project you need to define what does the client want, you look at costs, their expectations, the quality

requirements and you keep them in the loop and you can align your plans accordingly throughout the process by improving where necessary to ensure that you achieve the deliverables that were expected”.

- *Do maintenance after implementation (A1, A2, A3):*

All three interviewees were in agreement that maintenance and regular upgrades/updates should be done after the project/system is released, implemented and handed over. Interviewee A1 stated that the business “might not have the skill and the capability”, which is the reason that knowledge transfer is so important. She furthermore emphasised that “the solution that is often on the table needs to be planned in such a way that it gives me the freedom to make sure we can frequently upgrade, we can frequently maintain and we are in the position to do that, and obviously the client must be involved in the process”.

- *Ensure people are accountable for their tasks (A2):*

The hybrid APMM (ver. 0) cannot ensure that the team members are accountable for their tasks, that is the project manager’s responsibility. Interviewee A2 stated that “the key thing is to keep the people accountable for what they need to do and for what they have agreed to do and to manage that with a tight hand and obviously on a regular basis”.

- *Formally close-out, sign-off and hand-over a project (A1):*

It is important to formally sign-off all deliverables completed, hand them over and close-out the project to ensure that the client does not later claim that certain expectations were not satisfied. Interviewee A1 explained that in her experience, projects that are not formally closed out are cause for concern because “there are some snags and some issues that are still residing from the project side that the client has inherited and does not have control over, so it is extremely important” to obtain a formal sign-off and acceptance from the client.

- *Inform people of changes and the project managers must ensure that the changes are enforced (A1):*

Interviewee A1 explained that in the large organisation, users and other stakeholders are not kept updated when changes are done on a project. In many cases, it was because the client base was not identified upfront or because there was a communication breakdown. Often changes are approved by stakeholders but they are not enforced because there is need for a firm professional project manager.

- *Integrated communication (A1, A3):*

Interviewees A1 and A3 stated more than once that communication is the fundamental component for any project to succeed. If there is a breakdown in communication, the possibility of delivering a successful project is drastically decreased. Interviewee A1 stated that the “key to integration is communication”.

- *Keep PMM/ASDM simple, minimise documentation (keep important documents) and focus on completing the work (A1, A2, A3):*

All three interviewees were in agreement that the PMM/ASDM must be kept as simple as possible in order to minimise documentation and keep team members focused on their assigned responsibilities. There was also agreement that some crucial documentation that guides the project cannot be eliminated. Interviewee A1 stated that their project managers “always complain about documentation, saying leave us to do the job, leave us to deliver and execute the project”. Furthermore, she explained that the application of a PMM/ASDM must not “be over done and I think keeping it simple and concise is the way to go, especially in a changing environment”.

- *Keep stakeholders involved and provide regular feedback – this will ensure easier sign-offs (A1, A2, A3):*

All three interviewees were in agreement that it is very important to involve stakeholders constantly and provide feedback to stakeholders to ensure that the project stays on track. Interviewee A1 stated that “in this day and age you cannot, especially in our organisation, execute anything without keeping your stakeholders informed on a day-to-day basis and to get their support”. Interviewee A3 stated that by “involving your clients and stakeholders it makes it easier for you to get those sign-offs”. When stakeholders feel comfortable with the progress of a project, and if they are informed of all aspects of the project on a regular basis, sign-offs will be obtained more quickly and without hassle because the stakeholders will know that some areas of a project might come to a standstill if the sign-off is not provided.

- *Know your stakeholders and team (A3):*

It important to realise that the project manager will gain much more from the project team and the stakeholders if he/she knows them at a deeper level than just being people with responsibilities. Interviewee A3 summarised interpersonal skill aptly: “you must know your people, what makes them tick what makes them sick what makes them glad and what makes them sad”.

- *Log and implement lessons learnt as the project progresses to prevent repeating mistakes (A1, A2, A3):*

All the interviewees were in agreement that lessons learnt as the project progresses must be implemented in the current, other and future projects to ensure that the same mistakes are not repeated. This will help projects not lose money and valuable time. Interviewee A1 stated that if you accept criticism from stakeholders and apply it to “see what lessons you can learn from their experiences it can add value to the organisation”. Interviewees A2 and A3 stated that “we can learn from others” and “what is the purpose if lessons learnt are only shared at the end of the project”.

- *Manage changing/unclear requirements – give stakeholders what they want, not what developers think they need (A2):*

Interviewee A2 stated that in a “fast-paced and changing environment, requirements change and they aren’t always clear”, which is the reason that they must be “recorded and tracked” regularly. Stakeholder requirements are not always clearly defined and they tend to change regularly, which is the reason it is important for the project manager to apply adequate change-management procedures to ensure that the project does not fail.

- *Need a strong project manager/leadership (A2, A3):*

The successful execution of a PMM/ASDM and the delivery of a project rest in the hands of the project manager. If the project has a strong project manager with good leadership skills, the possibility of delivering a project successfully is increased. Interviewees A2 and A3 commented that “you need a very strong project manager” and “number one for me is leadership”.

- *Need adequate funding/budget (A1):*

Funding is the foundation of any project, even if the project is done without the intention of making money from it. If there is no money, there will be no project. For this reason, funds must be available to satisfy stakeholder expectations, otherwise the stakeholder expectations must be adjusted to suit the budget available or if accepted the budget must be adjusted to suit new or changing stakeholder expectations. Interviewee A1 mentioned that if there are “constraints on the funding of a project”, it places pressure on the team to deliver something of poorer quality than what would be possible with adequate funding.

- *Need a PMM/SDM that is flexible and agile – with the ability to adapt to change (A1, A2, A3):*

All three interviewees were in agreement that there is a need for a PMM/SDM that is flexible and sufficiently agile to adapt to changes that might occur in the project and business environment. Interviewee A1 mentioned that in their organisation they require a PMM with the necessary “openness for immediate midstream changes that might occur”, because “I don’t think we have the level of flexibility to manage our changes and our midstream diversions very well”. Interviewee A2 stated that the level of “flexibility already needs to be factored into the project so that when you get to a certain stage you have not completely missed your client requirements”. Interviewee A3 stated that the PMM/SDM “needs to be agile and adaptable because you know things change”.

- *Need integrated project management and system development (A1, A3):*

The areas in project management (human resources, procurements, risk, etc.) must be integrated with each other to ensure that when an activity occurs in one area, it is communicated to the other areas because one change or risk can affect a project as a whole. Interviewees A1 and A3 summarised the need for integrated project management and system development by stating that “project management and the development of systems should be integrated into your day to day activities”, “integration is the key” and the

“biggest driver is an integrated approach”.

- *Need the right people/resources on a team that is not short staffed with the right skills, allocated correctly and working together with the same vision – will result in a higher probability of project success (A1, A2, A3):*

All three interviewees were in agreement that the correct people with the required skills should be appointed to the project. Any organisation's most valuable assets are its people because they execute the project's tasks. The people must also be equipped with the right tools and resources to execute their responsibilities to the best of their ability. The chance of delivering a project successfully is much higher if the right people with the right skills, equipped with the right tools, are assigned to do the work.

- *Regular sign-offs are important, especially in a changing environment (A1, A2, A3):*

All three interviewees were in agreement that regular sign-offs must be completed to ensure that the team and the stakeholders stay on track. Regular sign-offs will also ensure that stakeholders do not later complain that certain deliverables were not completed correctly. It will also ensure that changes that might occur are managed more effectively, as some deliverables already completed and signed off might be affected by these changes. The stakeholders can thus see for themselves that changes affect the project financially.

- *Require proper monitoring and control procedure (A1, A2):*

Interviewees A1 and A2 explained that the project team must monitor and control continually the deliverables being executed in order to ensure that deliverables are delivered within cost, time, quality and scope constraints. Interviewee A2 mentioned that is important “to continuously review what is being done in terms of what you set out to achieve and to determine whether that is still being achieved”.

- *Stakeholder mapping (A1):*

In order to manage stakeholders effectively, the project manager should set up a stakeholder map before the project commences and update it regularly. Interviewee A1 stated that one component that could be added to the hybrid APM (ver. 0) is a stakeholder map.

- *Undefined/unclear scope (at beginning) – results in a great deal of rework (A1, A2):*

Interviewees A1 and A2 explained that if a project does not have a clearly defined scope before a project starts, this would result in rework being done. Interviewee A2 stated that “it is always very important to as far as possible in the beginning of the project to have your scope and your requirements, outputs, deliverables defined very clearly”. Interviewee A1 commented that “because you don't have that clear scope, you don't have that clear direction yet”.

Theme 2: Environmental characteristics

- *Constantly fast-changing environment (dynamic) – change is the only constant (A1, A2, A3):*

Each interviewee in case study A mentioned more than three times that they work in a constant and fast-changing environment. More than once, the interviewees stated that “change is the only constant”. Interviewee A1 explained that the “environment is so ever changing and the nature and the types of projects are so ever changing” that they need to “adapt and you just expect that change is part of your everyday life”. She also stated that in most projects, they are “flying the aeroplane whilst you are still busy building” it.

- *Project environment is challenging (A1):*

The environments in which many projects need to be delivered in large organisations are very challenging and complex. Interviewee A1 stated that “it’s an environment where you are expected to perform miracles and people do perform miracles ... pick your battle and you fight your battles which are worth fighting”.

- *Stakeholders expect things to change immediately and they expect/force team members to deliver to unrealistic deadlines (A1, A2, A3):*

In large organisations, stakeholders are very demanding and request things change immediately, and if they do not change immediately, they tend to get very upset although they often do not want to understand even if the project manager explains it in detail more than once. Interviewee A2 commented that “in our environment you are ordered to deliver something on time when people want it” without them giving a guideline of what exactly they are looking for.

- *Users do not know what they want (A1):*

As soon as the project team provides the user with something they tend to want more, or they say that they had something else in mind. If the project manager asks what else they require they cannot even explain what they want because the environment changes so frequently that the users’ way of thinking and understanding changes constantly, which results in users not giving the project team clear direction of what is expected. Interviewee A1 said that one of the major problems in a large organisation is that “people tend to go back on their word and their initial way of thinking”.

Theme 3: People factors

- *Human, behavioural, political and cultural factors are difficult to manage (A1, A3):*

Managing cultural, religious and behavioural factors in a multi-cultural country or region is very difficult in a project environment in which the main concern is delivering a project. Interviewee A1 stated that “in our organisation you cannot custom fit a process to the behavioural aspects and the cultural aspects”, as they are “difficult things to manage”.

Furthermore, she stated that “politics is an art”; it cannot be managed effectively in a business environment.

- *Performance management – people must be rewarded for doing a good job (A3):*

Interviewee A3 explained that one of the problems in their environment is that “performance management has not been implemented; people are not being rewarded for delivering”. If the organisation looks after their people, the people will look after the organisation.

Theme 4: Project management and system development principles

- *Incremental and iterative planning and development (A1, A2, A3):*

All three interviewees agreed that breaking up a project into manageable increments and iteratively completing each increment is an effective way of managing a project, and it ensures that the project team is not overwhelmed by the scope of a large project. Interviewee A2 stated that “it is a good approach to break the work up into manageable pieces” because it “keeps it manageable in such a way that people don’t get overwhelmed” by the amount of work.

- *Rather speculate than plan because planning only eliminates some uncertainty – not all of it (A1, A2, A3):*

Planning can only be done based on certainty. As soon as the team has to predict when deliverables should be completed, they are not making decisions based on facts but assumptions, which are most likely to change. Planning can thus only eliminate some uncertainty, not all of it. Interviewee A1 stated that planning “does take some of the uncertainty away but I do not think all of it because things change”.

- *Continuously review and improve the deliverables (A2):*

The aim of any project team should be to deliver products of the best possible quality within the time and cost constraints provided to them. Interviewee A2 explained that “a continuous improvement process is needed to review what you’ve developed and implemented and see how you can improve it further”.

- *Practitioners do apply project management and system development principles (A2, A3):*

Interviewees A2 and A3 mentioned more than once that they apply project management and system development principles every day in order to manage their responsibilities. After asking whether they apply project management and system development principles, interviewee A3 stated “certainly we do, I mean developing systems and managing projects are part of our daily outputs basically”.

- *Practitioners with knowledge of PMBOK/PRINCE2 do not apply it theoretically/fully by the book, only partially, because practically as a whole they do not work (A1, A2, A3):*

During the interviews, all of the participants stated that they knew and applied some of the aspects of these PMMs. Furthermore, in all the interviewees, it was apparent that PMBOK

or PRINCE2, if applied, was only applied partially, never fully. The reason for this is that it does not cater for the unique nature and circumstances of projects in large organisations. Interviewee A1 said that “I would be lying if I say that I do apply it and that I live by it”. Interviewee A3 stated that “PMBOK has its advantages and so does PRINCE2 as well and I think in most cases it’s more appropriate but for us there is no single one, you know we need to integrate, to make a hybrid one or to let it work together”.

- *Practitioners do not follow formal change-management procedures to manage change (A3):*
Interviewee A3 stated that there is “no formal process or project process to manage change”. Furthermore, she explained that projects tend to run over the scope, time and cost estimates. This concern can be addressed by having a formal change-request procedure in place to log, monitor and manage changes in order to minimise the impact change has on projects.
- *There is no “best breed” PMM/ASDM (A1, A2, A3):*
All three interviewees were in agreement that there is no PMM/ASDM that can be applied to all types and sizes of projects because every project has a unique nature. Interviewee A2 summarised this by saying that “all projects differ in nature so you can use different kinds of methodologies to custom make a methodology for a specific project based on the deliverables to be delivered”.

Theme 5: Advantages of the hybrid agile project management methodology (ver. 0)

- *Adjusts to different types and sized projects (A1, A2, A3):*
All three interviewees in case study A agreed that the hybrid APMM (ver. 0) has the ability to adjust to projects that vary in size and type. Their reason for the hybrid APMM (ver. 0) adjusting to different sizes and types of projects is that “it has flexibility” (interview A1) and “it allows for the change” (interview A2).
- *Enforces a culture in which people can embrace change (A1):*
Interviewee A1 stated that the hybrid APMM (ver. 0) “enforces a culture where people can embrace change” in a constantly changing project environment.
- *Ensures a quick response and resolution of issues/risks (A1):*
Because clients, stakeholders and team members are constantly involved as the project progresses, there is a quick response and solution for every problem that arises. Interviewee A1 stated that by applying the hybrid APMM (ver. 0) “everybody would be involved on a day to day basis and if any significant issues arise on the project it can immediately be flagged”.
- *Ensures minimised and simplified documentation/processes/semantics and focuses on completing the work (A1, A2, A3):*
All three interviewees in case study A agreed that the hybrid APMM (ver. 0) addresses the

CPSF (see theme 1 above), of *keeping a PMM simple, minimise documentation (keep important documents) and focus on completing the work*. Interviewee A1 explained that in “an organisation where time is against you and you are almost flying the aeroplane whilst you are still busy building you have to have something simple, not get bogged down in too much paperwork, too much jargon, semantics, process and phases”. Interviewee A2 emphasised that is important “not to focus on too many documents and because given that our requirements change so often those obviously need to change also”. Interviewee A3 explained that by applying the hybrid APMM (ver. 0), it is “less exhaustive because there is less paperwork, you know you focus really on getting the job done, there are processes involved, processes I think is agile enough to let you focus on what needs to be done and not necessarily get bogged down with paperwork”.

- *Ensures that benefit realisation is completed (A1):*

The hybrid APMM (ver. 0) addresses the identified CPSF (see theme 1 above) of *doing benefit realisation regularly*. Interviewee A1 explained that “this process makes sure that from the onset the various stakeholders are involved, communicated and there is a structured way of monitoring and controlling in terms of deliverables that are set versus the close-out thereof”. Furthermore, she explained that one of the objectives when applying the hybrid

APMM (ver. 0) “is to ensure that what we build and deliver at the end of the day is as what was expected by the client”.

- *Ensures that continuous improvement takes place (A2):*

The hybrid APMM (ver. 0) allows for continuous improvement in the *iterative refactoring cycle* (see Figure 5.6).

- *Ensures that the responsibilities of a project manager are not taken away (A1):*

The project manager still has the responsibility of delivering and executing the project. In order to do this, a strong project manager with good leadership skills is required.

- *Is an integrated approach, enhances integration and ensures integrated communication (highlight: collaborative development phase; A1, A3):*

The hybrid APMM (ver. 0) addresses the CPSFs (see theme 1 above) of *integrated communication* and the *need for integrated project management*. Interviewee A1 stated that communication is embedded throughout the hybrid APMM (ver. 0). Furthermore, she emphasised that “the whole collaborative approach, integration, communication and stakeholder management are some of the areas that I think we can really benefit from – this does not give me the flavour of a ‘silo’ mentality”. Interviewee A3 explained that the hybrid APMM (ver. 0) is well designed because communication channels are open to stakeholders and it allows for integration and regular feedback.

- *Is a simple process (A1, A2, A3):*

All three interviewees in case study A agreed that the hybrid APMM (ver. 0) is simple and easy to apply in practice. Each interviewee stated more than four times that the approach is simple, relevant and easy to apply in practice.

- *Is acceptable, implementable, flexible and adaptable to its surrounding environment (A1, A2, A3):*

All three interviewees in case study A agreed that the hybrid APMM (ver. 0) is flexible and adaptable to the project and business environment. Each interviewee stated at least three times that the approach is flexible and adaptive.

- *Is a new and fresh approach (A1):*

Interviewee A1 explained that the manner in which the hybrid APMM (ver. 0) approaches project management is “new and fresh”.

- *Is not a traditional approach and has the possibility of performing better than traditional approaches (A1, A2, A3):*

All the interviewees were in agreement that the hybrid APMM (ver. 0) is not a traditional PMM and it can perform better than other PMMs, such as PRINCE2 and PMBOK. Interviewee A1 stated that “it’s not a traditional approach and it has the possibility to perform better than other”, while interviewee A2 explained that it is a more effective and simple approach in comparison with other PMMs.

- *Keeps client/stakeholder constantly involved and satisfied (A1, A2, A3):*

All the interviewees were in agreement that when the hybrid APMM (ver. 0) is applied it will ensure that stakeholders are constantly involved as the project progresses to ensure that their expectations are managed and satisfied. Each interviewee mentioned at least three times that the hybrid APMM (ver. 0) ensures stakeholder involvement. Interviewee A1 stated that the hybrid APMM (ver. 0) “makes sure that from the onset the various stakeholders are involved” and that there “is a very strong focus on stakeholder management”. Interviewee A2 explained that client engagements happen continually when using the hybrid APMM (ver. 0). Interviewee A3 emphasised that “the stand-up meetings, the feedback meetings, the regular, whole integration of the processes, speaking with clients or stakeholders, keeping them involved is adding a lot of value to the whole process”.

- *Monitors and controls deliverables (A1):*

The hybrid APMM (ver. 0) is designed to monitor and control every deliverable as the project progresses to ensure that what was promised is delivered. Interviewee A1 stated that the hybrid APMM (ver. 0) provides a “structured way of monitoring and control in terms of deliverables”.

- *Provides a suitable change-request form to manage change (A1, A2, A3):*

All the interviewees were in agreement that the change-request form provided is suitable for

managing, controlling and assessing the impact of project changes.

- *Will assist project managers and teams to deliver projects successfully in constantly changing project environments (A1, A2, A3):*

All the interviewees in case study A agreed that the hybrid APMM (ver. 0) will assist project managers and team members in delivering projects successfully in a constantly changing project and business environment. This emphasises one of the advantages of the hybrid APMM (ver. 0).

- *Will deliver project within time, cost, and quality constraints (A3):*

The hybrid APMM (ver. 0) addresses the identified project management and system development principle (see theme 4 above) of *using the right people and tools to deliver within time, cost, quality and scope constraints*. Interviewee A3 explained that by applying the hybrid APMM (ver. 0) in their project environment it will ensure that projects will be delivered within time, cost and scope constraints because the deliverables are continually monitored, controlled and reported on.

- *Allows speculation on the unknowns and planning on the facts (A1):*

The hybrid APMM (ver. 0) addresses the identified project management and system development principle (see theme 4 above) *rather speculate than plan because planning only eliminates some uncertainty – not all of it*. Interviewee A1 stated that “that is the beauty of it, you will take some initial uncertainty away through your planning, and through your speculation and your forecasting” the project manager can estimate the unknown by using the experience of the people involved in the project.

- *Allows the team to achieve its objectives more quickly (A2, A3):*

Interviewees A2 and A3 explained that the hybrid APMM (ver. 0) will ensure that the project manager and team members complete deliverables more quickly than normal. Interviewee A2 stated that the hybrid APMM’s (ver. 0) “focus is more on achieving objectives quickly” and if “changes occur, you get solutions easier and more quickly”. Like interviewee A2, interviewee A3 emphasised that the hybrid APMM (ver. 0) “is really focused on project delivery and, getting things done”.

Theme 6: Shortcomings of the hybrid agile project management methodology (ver. 0)

The following observations and possible improvements were mentioned in the interviews.

- *Needs to include a stakeholder mapping (A1):*

Interviewee A1 mentioned that it would be beneficial if a stakeholder mapping were included in the hybrid APMM (ver. 0).

- *Requires a project manager who enforces and manages the process effectively (A1):*

The role of an effective and strong project manager with good interpersonal and leadership skills was emphasised by interviewee A1. Furthermore, she stated that “all of it will not bring

you anywhere if the management thereof and the enforcement thereof is not done correctly” because the hybrid APMM (ver. 0) itself is not flawed.

6.2.1.2 Results of case study B (small organisation)

Case study B was executed as explained in Chapter 4. The results of the analysis of the content of interviews B1, B2 and B3 are presented under the six themes.

Theme 1: Critical project success factors

- *Consider the nature of a project before applying a PMM/ASDM (B2, B3):*

Every project has a unique nature. Both interviewees B2 and B3 explained that the project environment must first be assessed before a PMM/ASDM is applied to ensure that the PMM/ASDM suits the environment. Interviewee B2 stated that the project manager must first “validate the scope and surroundings and then decide what will be feasible”.

- *Do benefit realisation regularly (B1, B2, B3):*

All the interviewees in case study B identified the importance of delivering what was promised in the business case. Interviewee B1 stated that “every single project team member must make a commitment to deliver what is being asked upfront and if that cannot be the case with the information at hand then that honest and mature communication needs to take place because it will result in scope creep”. Interviewee B3 commented that “you have to stick with what was the requirement in the beginning, and both parties need to agree to that, and any changes need to go through a formal change control process”.

- *Do maintenance after implementation (B1, B2, B3):*

All the participants in case study B stated that regular updates and maintenance after the project has been delivered is important. Interviewee B1 stated “Yes very much so and I think that most of the failures that we’ve seen in projects are where updates and maintenance for instance is underestimated, specifically teams would not budget for it”. Interviewee B2 explained that “to me the implementation and the bedding down is something that has to be considered, it has to be part of the project plan and it has to be agreed upfront and it’s something that cannot be skipped over. I think it’s massively important”. Interviewee B3 stated that “I definitely think it’s important, you will run into situations where you delivered a project or product that is only usable for a certain amount of time until it needs to be upgraded or improved”.

- *Formally close-out, sign-off and hand-over a project (B2, B3):*

Interviewees B2 and B3 viewed a formal close-out and hand-over as important. Interviewee B2 summarised this by explaining that “it’s critical, especially as consultants we need to do a proper hand-over to our clients and the permanent staff that is involved in the project and then also the people that needs to use it needs to understand exactly what the capabilities

are”.

- *Integrated communication (B1, B2):*

Interviewees B1 and B2 viewed communication as integral to the various aspects and areas of a project. Interviewee B1 stated that “to me project management is really about communication, if we have strong communication we’re happy ... strong communication is something that needs to be embedded culturally as well”. Culturally, communication can be an obstacle, especially in Africa where there are so many different cultures and languages. It is important to understand the different cultures of the people involved in a project to ensure that communication is not offensive. Interviewee B2 stated that “it’s critically important to have good communication”.

- *Keep PMM/ASDM simple, minimise documentation (keep important documents) and focus on completing the work (B1, B2, B3):*

All the interviewees in case study B were in agreement that it is important to keep it simple and to focus on completing the work. Interviewee B1 answered by stating: “keeping project management and system development methodologies simple, yes, minimise documentation, absolutely not, and focus on completing the work, yes”. The reason for him saying that documentation should not be minimised is because “some of our clients are dependent on proper documentation”. Furthermore, he stated that “I absolutely believe in minimising documentation for the sake of removing worthless documentation because often projects generate huge amounts of red tape which really has no purpose”. Interviewee B2 explained that it is important for the “person that needs to do the actual work not to sit and do documentation on a daily basis because you are actually wasting time on the project itself”.

- *Keep stakeholders involved and provide regular feedback – this will ensure easier sign-offs (B1, B2):*

Interviewees B1 and B2 explained that the stakeholders must be kept involved by providing regular feedback on the status, progress, risks and issues of the project. Interviewee B2 provided a good explanation by stating that this is “why the ongoing meetings and updates is a necessity in any project to basically keep everyone up to date. I’ve also said that I think ongoing sign-off is critical because then everyone is up to date and they do know what is expected of all the parties involved in projects so I would say definitely an ongoing update and feedback meetings to higher management and stakeholder is critical”.

- *Log and implement lessons learnt as the project progresses to prevent repeating mistakes (B1, B2, B3):*

All the interviewees in case study B were in agreement that lessons learnt must be implemented as the project progresses to ensure that the same mistakes are not repeated. Interviewee B1 stated “definitely as the project progresses”, while Interviewee B2 said that if

lessons are implemented practitioners “can actually start learning in the current increment and use those lessons to plan for following increments” to ensure that the same kinds of mistakes are not made more than once. Interviewee B3 explained that “it is better to log lessons throughout the project and not just at the end” of the project.

- *Manage changing/unclear requirements – give stakeholders what they want, not what developers think they need (B1, B2, B3):*

All the interviewees in case study B emphasised that the requirements they gather in their environment are in most cases not clear and tend to change regularly. Interviewee B1 explained that is important not to give stakeholders what the developers think they need, the team must understand the stakeholders’ expectations and deliver on those. Interviewee B2 stated that on a regular basis there are “new requirements and changing requirements that you need to understand and check what impact it will have on your current project deadlines”. Interviewee B3 stated that “the business necessitates change and change is always necessary for a business to keep up with its competitors, so keeping up with changes is inevitable”.

- *Need a strong project manager/leadership (B1, B2, B3):*

All the interviewees in case study B were in agreement that a project requires a solid project manager who can guide the project to successful completion. Interviewee B1 explained that a project requires “a real effective project manager, someone who has come through the ranks, who understands the complexities, who understands top level decision-making”. Interviewee B3 stated that a project needs a “project manager that understands the business and one that can pull off a successful project”. Interviewee B3 explained that team members have great comfort and a definite perceived value of a project manager who can assume responsibility.

- *Need a PMM/SDM that is flexible and agile – with the ability to adapt to change (B1, B3):*

Interviewees B1 and B3 agreed that they require a PMM/SDM that is flexible and adaptable to its project environment. Interviewee A1 stated that “we have to be more nimble ... and to be able to do that my expectation of every single member within my organisation is to have an understanding of what it is that they need to do, to be absolutely a specialist in their job” and to adapt when necessary. Interviewee B3 explained that organisations have to change and adapt to different project environments in order to be a competing business where change occurs daily.

- *Need integrated project management and system development (B2):*

Interviewee B2 agreed that the different areas of project management and system development must be integrated with each other and with the rest of the business and that there is a need in their business environment for a PMM/ASDM that does this effectively.

- *Need the right people/resources on a team that is not short staffed with the right skills,*

allocated correctly and working together with the same vision – will result in a higher probability of project success (B1, B2, B3):

All the interviewees in case study B agreed that a project requires the correct people with a united vision to complete a project successfully. Interviewee B1 believed that “not any methodology can ever substitute competent team members” and explained that “you’ve got to have the right people and you’ve got to have the right tools and you’ve got to have sufficient capital”. Furthermore, he emphasised that if “you have an unrealistic target and if you have the wrong team you’ve got no chance to deliver”. In addition, he explained that short-staffed teams would lead to project failure.

- *Project manager and team members must be on board to utilise and enforce a PMM (B1):*
Interviewee B1 stated that “I have seen it over and over; you cannot look towards one person to enforce a project management methodology it has to be within the team ... the only way to do that is to have all hands on deck with a project management approach”. It is not only the project manager’s responsibility to enforce a PMM, it is the responsibility of every team member to understand and execute the different aspects of a PMM, which will make it successful.
- *Regular sign-offs are important, especially in a changing environment (B2):*
Interviewee B2 viewed regular sign-offs as very important because this ensures that the stakeholders and team members are kept up to date regarding what has been completed and what still needs to be completed. In contrast, interviewees B1 and B3 agreed that there must not be too many sign-offs as this may result in project teams being counterproductive while they await sign-offs. This will be discussed in theme 6.

Theme 2: Environmental characteristics

- *Constantly fast-changing environment (dynamic) – change is the only constant (B1, B2, B3):*
All the interviewees in case study B described their working environment as dynamic and changing almost on a daily basis. Interviewee B1 stated that “we cannot be rigid in the environments we function”, and Interviewee B3 explained that “we have projects coming from a changing business”.
- *Stakeholders expect things to change immediately and they expect/force team members to deliver to unrealistic deadlines (B1, B3):*
Interviewees B1 and B3 agreed that stakeholders sometimes have unrealistic expectations and deadlines. Interviewee B1 emphasised that the “biggest reason for projects to fail is ridiculous upfront expectations”. Interviewee B3 mentioned that one of the reasons that projects fail is because “timelines are sometimes too tight that are unreasonable”.
- *Team members feel that the project manager does not always understand (B3):*

Interviewee B3 stated that “there is a general tendency that team members usually feel that project managers do not always understand the intricacies of a project or the detail of how the business functions ... they also don’t understand in detail what the project team talks about”. This a unique factor that only surfaced in case study B.

Theme 3: People factors

- *Human, behavioural, political and cultural factors are difficult to manage (B1):*

Interviewee B1 agreed that cultural, political and human behaviour is complicated to manage and stated that the best solution to the problem is to manage client expectations instead of different cultures. Furthermore, he explained that “we have one culture and that is one of delivery ... don’t get me wrong it’s not like we don’t have an appreciation for different cultures but there is no management of culture because we all have to commit to deliverables. We appreciate culture and we understand that different people operate differently.”

- *People talk too much and do too little (B3):*

Interviewee B3 mentioned that in their client environment, people “tend to talk too much and do too little”, especially in meetings. There is a general tendency in organisations to hold too many meetings at which everyone wishes to make his/her voice heard instead of focusing on completing the work.

- *Performance management – people must be rewarded for doing a good job (B1):*

Interviewee B1 explained that “you need to take a long hard look in the mirror and say why their efforts weren’t really up to speed” and the reason normally is that they have never been rewarded for the months even years of hard work they have done. People need appreciation in order to perform, they need to know that what they are doing is of value – it will give them a sense of purpose.

Theme 4: Project management and system development principles

- *Feedback meetings have to be purpose driven and the right people must be invited to these meetings (B2):*

Much time is wasted in organisations because of meetings that do not have a specific purpose and the wrong people being invited, which results in unnecessary conversations that are not relevant to the project or the deliverables that must be completed. Furthermore, Interviewee B2 stated that “in feedback meetings too many people are involved and too little work is done because it’s not practical to set up meetings like these”.

- *Incremental and iterative planning and development (B1, B2, B3):*

All the interviewees in case study B agreed that the work to be completed must be broken down into increments, where each increment is developed iteratively. Interviewee B1 stated

that “what we find in practice is that the deeper you get into the detail, the more you have to break the project down and you almost need to be nimble, agile is the word and you need to stay agile ... you have to have an iterative approach because I believe bite-sized pieces of work works typically well ...”. Interviewee B2 explained that incremental and iterative development ensures that the project is completed quicker than “doing the whole project as one major phase, because it is easier to control and maintain pieces than trying to do the whole project as one main big increment”. He furthermore stated that the project team must be cautious not to “divide it into too many increments” because then the management of the different increments running simultaneously might become a problem. Interviewee B3 explained that incremental and iterative development “assist team members to approach a project systematically”.

- *Rather speculate than plan because planning only eliminates some uncertainty – not all of it (B1, B2, B3):*

All the interviewees in case study B were in agreement that it is better to speculate when there are uncertainties and to do planning only on definite facts. Interviewee A2 stated that “you can plan up to a certain point, but there’s always certain factors and aspects of projects that you won’t know what the outcome will be”. Interviewee B3 explained that “you can plan to a certain extent but there will always be uncertainties which you cannot account for in the beginning which will pop up throughout the duration of the project” then you need to start to predict or speculate what could happen.

- *Continuously review and improve the deliverables (B2):*

Interviewee B2 emphasised that it is important for the project team to review and improve continuously the deliverables they complete to ensure that a product of quality is delivered according to the time and cost expectations of the stakeholders.

- *Practitioners do apply project management and system development principles (B1, B2, B3):*

All the interviewees in case study B stated that they apply project management and system development principles in some or other form in their normal working day.

- *Practitioners with knowledge of PMBOK/PRINCE2 do not apply it theoretically/fully by the book, only partially, because practically as a whole they do not work (B1, B2, B3):*

All the interviewees in case study B were in agreement that while they know of PMBOK, PRINCE2 and other traditional methods, they do not apply these in full because they need a PMM that can adapt to the different types of projects they deliver. Interviewee B1 explained that “we do not apply rigid project management, in our organisation we focus on delivering complex projects and to deliver complex projects you can’t really be too rigid in your approach”. Furthermore, he stated that “PMBOK and PRINCE2 does have value, but I believe as methodologies they are flawed and in terms of implementation I’ve not seen the

best of it ... you have to have an iterative approach that is why I find PMBOK and PRINCE2 naïve ... PMBOK openly says that you can with general principles deliver any project and I do not believe that – I believe that it is false”.

- *There is no “best breed” PMM/ASDM (B1, B2, B3):*

All the interviewees in case study B agreed that there is no “best breed” PMM or ASDM. According to interviewee B2, the PMM or ASDM a project team applies is dependable on “what kind of project you are working with”.

Theme 5: Advantages of the hybrid agile project management methodology (ver. 0)

- *Adjusts to different types and sized projects (B1, B2, B3):*

All the interviewees in case study B agreed that the proposed hybrid APMM (ver. 0) can be applied to projects of different sizes and types. Interviewee B3 explained that “whether it’s a big or a small one, the fact that you can change as you go and apply it to different scenarios and sizes of projects would be something special in our environment”.

- *Ensures a quick response and resolution of issues/risks (B3):*

Interviewee B3 agreed that the hybrid APMM (ver. 0) ensures that issues are quickly resolved by the way that it is constructed. Interviewee B3 stated that “the adapt, direct, monitor and control function that sits in the middle of the whole refactoring process encompasses every stage of the methodology and gives the methodology the ability to obviously adapt and resolve issues when they occur”.

- *Ensures minimised and simplified documentation/processes/semantics and focuses on completing the work (B3):*

Interviewee B3 agreed that the hybrid APMM (ver. 0) focuses on completing the work and ensures that documentation is minimised and simplified. Interviewee B3 explained that the hybrid APMM (ver. 0) gives him “the feeling that it is simple, to the point, not time consuming and it seems like less documentation will be done, which is always welcome”.

- *Ensures that continuous improvement takes place (B1, B2, B3):*

All the interviewees in case study B agreed that the hybrid APMM (ver. 0) ensures that deliverables will be continually improved as the project progresses. Interviewee B2 summarised this by stating that the hybrid APMM (ver. 0) “re-evaluates the deliverables completed in the process to see if you should improved the deliverables or the general project as such. If you need to go back to the drawing board make sure that your new changing requirements have been added” to improve the overall design.

- *Ensures that lessons learnt are implemented (B2):*

Interviewee B2 stated that one of the advantages of the hybrid APMM (ver. 0) is that “that you can learn from previous mistakes and shortfalls and from previous projects to ensure that they don’t affect current projects and future projects” in the organisation. Lessons learnt

and solutions found for problems that resulted in the project losing valuable time and money must be implemented timeously to ensure that the same problems do not cause further difficulties in the organisation's projects.

- *Focuses on incremental and iterative development (B1, B2, B3):*

All the interviewees in case study B were in agreement that the hybrid APMM (ver. 0) has the advantage of breaking the project work down into increments, where each increment is developed iteratively. Interviewee B1 explained that the main advantage of the hybrid APMM (ver. 0) is "the incremental and iterative nature". Interviewee B2 stated that the hybrid APMM (ver. 0) has more "control over the project because it's divided into increments". Interviewee B3 explained that what stood out for him is "the incremental release plan approach as a practical way to do a breakdown of all the project work".

- *Is an integrated approach, enhances integration and ensures integrated communication (highlight: collaborative development phase; B1, B2):*

Interviewees B1 and B2 viewed the hybrid APMM (ver. 0) as an integrated approach that enhances integrated communication across different areas of the project and the business as a whole. Interviewee B1 summarised this by stating that "what is unique about it is the fact that it integrates so many ideas ... this integrated approach is quite new and fresh".

- *Is a simple process (B2, B3):*

Interviewees B2 and B3 stated the hybrid APMM (ver. 0) is "a simple process" that can be understood and applied by any person if the time is taken to study it in detail.

- *Is acceptable, implementable, flexible and adaptable to its surrounding environment (B1, B3):*

Interviewees B1 and B3 were in agreement that the hybrid APMM (ver. 0) is flexible and can adapt to different project environments. Interviewee B1 stated that he believes the hybrid

APMM (ver. 0) is "very adaptable". Interviewee B3 explained that the hybrid APMM (ver. 0) "gives the project the ability to obviously adapt". As stated in one of the previous advantages, interviewee B3 also commented that "whether it's a big or a small one, the fact that you can change as you go and apply it to different scenarios and sizes of projects would be something special in our environment".

- *Is a new and fresh approach (B3):*

Interviewee B3 views the hybrid APMM (ver. 0) as a new and fresh approach, stating that "it's quite a fresh approach for me, which would make especially team members happy".

- *Is not a traditional approach and has the possibility of performing better than traditional approaches (B1, B3):*

Both interviewees B1 and B3 stated that they think the hybrid APMM (ver. 0) will perform better than traditional PMMs. Interviewee B1 stated that "Well it certainly describes the way

that we work better than a traditional approach and accordingly I would have to say it is better”.

- *Keeps client/stakeholder constantly involved and satisfied (B1, B2, B3):*

All the interviewees in case study B were in agreement that the hybrid APMM (ver. 0) focuses on keeping the stakeholders satisfied and involved on a regular basis. Interviewee A1 stated that the hybrid APMM (ver. 0) “makes sure that people are and stays on board ... it forces them to be more closely involved for more feedback and decision-making”. Interviewee B2 stated that the regular feedback and constant client involvement will ensure that “everyone is on the same page”.

- *Provides clear direction and vision (B2):*

When interviewee B2 referred to the “vision and direction” of a project, the interviewer asked whether the hybrid APMM (ver. 0) provides vision and direction, to which the participant replied “yes”.

- *Provides a suitable change-request form to manage change (B1, B2, B3):*

All the interviewees in case study B were in agreement that the change-request form provided is a suitable means of managing, controlling and assessing the impact of project changes. Interviewee B1 stated that “this is definitely sufficient”. Interviewee B2 commented that the change-request form “looks adequate, very adequate”.

- *Will assist project managers and teams to deliver projects successfully in constantly changing project environments (B2, B3):*

Interviewees B2 and B3 both agreed that the hybrid APMM (ver. 0) will assist project managers in delivering projects in an ever-changing project environment.

- *Assists in managing uncertainties (B2, B3):*

Both interviewees B2 and B3 explained that the hybrid APMM (ver. 0) will assist project managers in managing uncertainties more efficiently and more effectively. Interviewee B2 stated that by applying the hybrid APMM (ver. 0) “you will be able handle these kind of uncertainties in the future”. Interviewee B3 explained that the hybrid APMM (ver. 0) “places emphasis on identifying as much uncertainties as possible beforehand and then it makes proper room for those uncertainties to be managed”.

Theme 6: Shortcomings of the hybrid agile project management methodology (ver. 0)

- *Must be careful not to discuss the same thing repeatedly (B3):*

Interviewee B3 gave “a word of warning, although the approach is to make it as simple as possible people must not waste time on unnecessary discussions” especially during the *improve and adapt* step in the iterative refactoring cycle.

- *Must not give team members free rein – need critical decision-makers (B1):*

Interviewee B1 stated that it is important for a PMM or ASDM not to “give the project team

carte blanche” to do what they want within a specific increment and log change requests. There needs to be urgency amongst team members to complete deliverables and increments in time and according to budget, even though there might be small changes. Change requests cannot be logged for the smallest changes that occur, as this will increase documentation.

- *Regular sign-offs are counterproductive (B1, B3):*

Interviewee B1 stated that there must be a balance because “regular sign-offs is a pain – to have someone run around with a document to try and find someone who is in five hundred meetings for the whole week is very counterproductive and I have seen that it breaks down the morale of a project team and often people that must sign are on leave”. Furthermore, he explained that sign-offs are important but that “you need big sign-offs” that contain more than one deliverable, especially in small-scale projects. Interviewee B3 stated that “I don’t think there should be too many sign-offs during the project timeline ... one in the middle and one in the end could do well for managing expectations and to make sure that stakeholders and the sponsors are involved.” In small organisations, it would be wise to rather have a minimal amount of sign-offs that are valuable than have sign-offs on a regular basis that cause frustration without it being of value.

6.2.2 Cross-case analysis

Following the grouping of the propositions of each case study under the six themes, the case studies will be compared using cross-case analysis as analysis method to determine similarities and differences between the propositions of the case studies as explained in Section 4.3.2.1. The aim was to determine whether the propositions that supported the interviews (A1, A2 and A3) of case study A were supported by the propositions that supported the interviews (B1, B2 and B3) of case study B. Another aim was to detect differences and similarities arising from the size of the organisation for each proposition.

6.2.2.1 Results of cross-case analysis of case studies A and B

The different propositions of the different interviews of case studies A and B are summarised under the six themes below. The interviews (A1, A2, A3, B1, B2 and B3) that supported a specific proposition are placed in brackets behind the proposition. If interviews for both case study A and case study B were placed behind the same proposition, this indicates that this proposition is relevant to large and small organisations. If interviews A1, A2 or A3 are the only interviews placed in brackets, this indicates that this proposition is only relevant to large organisations. If interviews B1, B2 or B3 are the only interviews placed in brackets, this indicates that this proposition is only relevant to small organisations. The propositions under each theme are grouped based on which propositions supported both large and small

organisations, and those that supported either only a large or only a small organisation. It is important to remember that interviewees A1 and B1 are senior management, interviewees A2 and B2 are project managers, and interviewees A3 and B3 are business analysts/programmers part of the project team.

Theme 1: Critical project success factors

The CPSFs that were supported by both large and small organisations are:

- *Consider the nature of a project before applying a PMM/ASDM (A2, A3, B2, B3):*
Both case studies supported the CPSF that the nature of a project must first be considered before a PMM/ASDM is applied to ensure that the PMM/ASDM has a better likelihood of delivering the project or system successfully.
- *Do benefit realisation regularly (A1, A2, A3, B1, B2, B3):*
All the participants of both case studies strongly supported that regular benefit realisation must be done to ensure that what was promised at the beginning of the project is delivered when the project is completed – taking into consideration that changes and unforeseen circumstances might have changed some of the original expectations and requirements.
- *Do maintenance after implementation (A1, A2, A3, B1, B2, B3):*
All interviewees in both case studies strongly supported that maintenance and regular upgrades must be done after the project has been deployed and implemented.
- *Formally close-out, sign-off and hand-over a project (A1, B2, B3):*
Both case studies support the formal close-out, hand-over and sign-off a project.
- *Integrated communication (A1, A3, B1, B2):*
Case studies A and B agreed that integrated communication is a CPSF that ensures effective integration amongst the different aspects and areas of a project, especially when team members work in different departments of an organisation.
- *Keep PMM/ASDM simple, minimise documentation (keep important documents) and focus on completing the work (A1, A2, A3, B1, B2, B3):*
All the interviewees in both case studies strongly supported that a PMM/ASDM must be kept simple and that documentation must be minimised. Minimising documentation does not mean that the project team must compile as little documentation as possible because documentation is important and project teams view it as the critical source to which to refer regarding what must be delivered by when and the manner in which it must be delivered. Important project documentation must still be completed, only documents that have no value to the purpose of the project are removed. Furthermore, all the participants of both case studies strongly supported that everyone involved in the project must focus on completing the work, instead of focusing on aspects that are not relevant to the project or that might cause the project to be delayed.

- *Keep stakeholders involved and provide regular feedback – this will ensure easier sign-offs (A1, A2, A3, B1, B2):*

Both case studies were in agreement that stakeholders must be kept involved by providing regular feedback, as this will ensure quick and easy sign-offs.

- *Log and implement lessons learnt as the project progresses to prevent repeating mistakes (A1, A2, A3, B1, B2, B3):*

Both case studies were in agreement that lessons must be logged and implemented as the project progresses to ensure that the same mistakes are not repeated.

- *Manage changing/unclear requirements – give stakeholders what they want, not what developers think they need (A2, B1, B2, B3):*

Case studies A and B agreed that changing/unclear requirements must be managed effectively to ensure that they do not cause scope creep or even the project to fail. It is important that stakeholder expectations and requirements be clearly defined to ensure that the project team knows what needs to be delivered and does not provide the stakeholders with what they think should be delivered.

- *Need a strong project manager/leadership (A2, A3, B1, B2, B3):*

Both case studies were in agreement that a strong project manager with good leadership skills is required to deliver a project in a challenging IT environment.

- *Need a PMM/SDM that is flexible and agile – with the ability to adapt to change (A1, A2, A3, B1, B3):*

Both case studies strongly support that an agile and flexible PMM/SDM is required that can adapt to change in order to deliver projects in an evolving project environment.

- *Need integrated project management and system development (A1, A3, B2):*

Both case studies were in agreement that an organisation requires project management and system development to be integrated with all other areas in the business to ensure transparency of all the management components required to deliver a project successfully. This is required especially where different departments (procurements, human resources, risk, etc.) are involved in delivering the same project.

- *Need the right people/resources on a team that is not short staffed with the right skills, allocated correctly and working together with the same vision – will result in a higher probability of project success (A1, A2, A3, B1, B2, B3):*

This CPSF was strongly supported and viewed as very important by both case studies. The most valuable resource of any organisation is its people, which is the reason that an organisation requires good staff with the required skills, allocated correctly and working towards the same goal. It is important that projects are not short staffed because it will result in the projects being delivered late, and the possibility of failure will be higher.

- *Regular sign-offs are important, especially in a changing environment (A1, A2, A3, B2):*

Both case studies were in agreement that it is important for the project team to obtain regular sign-offs, especially in a changing project and business environment, in order to ensure that control of the project is maintained.

The CPSFs that were supported only by the large organisation are:

- *Require proper monitoring and control procedure (A1, A2):*

This CPSF was supported by the senior manager and project manager of the large organisation, who stated that the proper monitoring and controlling procedures are required to ensure that a project is well controlled as it progress.

- *Stakeholder mapping (A1):*

The senior manager of the large organisation was the only participant that suggested the use of a stakeholder mapping to define stakeholders and reporting lines clearly.

- *Undefined/unclear scope (at beginning) – results in a great deal of rework (A1, A2):*

The project manager and senior manager of the large organisation agreed that the project scope must be clearly defined; otherwise, it will result in a great deal of rework as the project progresses.

- *Deliver quickly (A1):*

The senior manager of the large organisation was the only participant that explained the importance of team members in delivering quickly.

- *Inform people of changes and the project managers must ensure that the changes are enforced (A1):*

The senior manager was the only participant that mentioned the importance of people being informed of changes to ensure that they are not left behind, especially if the change will affect them when the project is deployed. She furthermore explained that it is the project manager's responsibility to enforce and ensure that approved changes are completed and communicated.

- *Ensure people are accountable for their tasks (A2):*

The project manager from the large organisation was the only participant who stated the importance of a project manager to ensure that team members are accountable for their tasks by doing regular follow-ups.

- *Know your stakeholders and team (A3):*

The team member from the large organisation commented that the team and project manager must know each other and the stakeholders also at a personal level. This will result in better management and control of team members and stakeholders, especially if behavioural and cultural factors are involved

- *Need adequate funding/budget (A1):*

The senior manager of the large organisation explained that money is the most important

driver of their projects, which is the reason that there is a need for adequate funding before a project commences.

The CPSF that was supported only by the small organisation is:

- *Project manager and team members must be on board to utilise and enforce a PMM (B1):*
The senior manager of the small organisation was the only participant that explained the importance of a whole team's involvement in enforcing and effectively utilising a PMM.

Theme 2: Environmental characteristics

The environmental characteristics that were supported by both the large and small organisations are:

- *Constantly fast-changing environment (dynamic) – change is the only constant (A1, A2, A3, B1, B2, B3):*
Both case studies agreed that they work in a dynamic and rapidly changing project and business environment.
- *Stakeholders expect things to change immediately and they expect/force team members to deliver to unrealistic deadlines (A1, A2, A3, B1, B3):*
Both case studies emphasised and strongly supported the concern that stakeholders sometimes think that they can dictate and change expectations and requirements when they see fit, which causes team members to have to deliver to unrealistic deadlines.

The environmental factors that were supported only by the large organisation are:

- *Project environment is challenging (A1):*
The senior manager of the large organisation was the only participant that stated that their project environment is very challenging.
- *Users do not know what they want (A1):*
The senior manager was the only participant from the large organisation that explained that users and clients do not know what they want. As soon they get something, either it is changed or they state that it is not what they wanted. If they are asked what is it that their require, they are unable to give a precise answer.

The environmental factor that was supported only by the small organisation is:

- *Team members feel that the project manager does not always understand (B3):*
The team member of the small organisation raised his concern that project managers do not understand the detail of deliverables and place the team members under tight deadlines without them considering the amount of work that needs to be done.

Theme 3: People factors

The people factors that were supported by both the large and small organisation are:

- *Human, behavioural, political and cultural factors are difficult to manage (A1, A3, B1):*

Both organisations were in agreement that behavioural, political and cultural factors that are present in a project are very difficult to manage.

- *Performance management – people must be rewarded for doing a good job (A3, B1):*

Both case studies agreed that people must be rewarded for good performance, as it keeps them motivated and it gives them a sense of purpose by knowing that they are bringing value to the organisation and that their work is viewed as important for the success of the project and organisation.

The people factor that was supported only by the small organisation is:

- *People talk too much and do too little (B3):*

The team member of the small organisation was the only participant who explained that people are not as productive as they should be because they talk too much about other subjects instead of doing the actual work.

There were no people factors that were supported by the large organisation.

Theme 4: Project management and system development principles

The project management and system development principles that were supported by both the large and small organisation are:

- *Incremental and iterative planning and development (A1, A2, A3, B1, B2, B3):*

Both case studies strongly supported the principle of doing incremental and iterative planning and development, where the project is broken down into increments and each increment is developed iteratively.

- *Rather speculate than plan because planning only eliminates some uncertainty – not all of it (A1, A2, A3, B1, B2, B3):*

Both organisations strongly agreed that team members should rather speculate than plan because planning is based on known facts, while a person speculates when uncertainties are involved.

- *Continuously review and improve the deliverables (A2, B2):*

Both case studies agreed that project deliverables should be continuously reviewed and improved.

- *Practitioners do apply project management and system development principles (A2, A3, B1, B2, B3):*

It was supported by both case studies that practitioners do apply project management and

system development principles in their daily jobs.

- *Practitioners with knowledge of PMBOK/PRINCE2 do not apply it theoretically/fully by the book, only partially, because practically as a whole they do not work (A1, A2, A3, B1, B2, B3):*

Both organisations were in strong agreement that although practitioners have knowledge of PMBOK and PRINCE2 they do not apply it by the book because practically as a whole they do not work. If practitioners apply PMBOK or PRINCE2, they only apply it partially.

- *There is no “best breed” PMM/ASDM (A1, A2, A3, B1, B2, B3):*

All the interviewees of both case studies strongly agreed there is no “best breed” PMM or ASDM that can be applied to all types of projects.

The project management and system development principle that was supported only by the large organisation is:

- *Practitioners do not follow formal change-management procedures to manage change (A3):*

The team member of the large organisation was the only participant who mentioned the fact that they do not formally apply a change-management procedure to manage change in their project or business environment.

The project management and system development principle that was supported only by the small organisation is:

- *Feedback meetings have to be purpose driven and the right people must be invited to these meetings (B2):*

The project manager of the small organisation commented that feedback sessions must be purpose driven and the correct people must be invited, otherwise time will be wasted on discussions not relevant to the project.

Theme 5: Advantages of the hybrid agile project management methodology (ver. 0)

The advantages of the hybrid APMM (ver. 0) that were supported by both the large and small organisation are:

- *Adjusts to different types and sized projects (A1, A2, A3, B1, B2, B3):*

Both the case studies were in strong agreement that the hybrid APMM (ver. 0) has the ability to adjust or adapt to different types and size projects.

- *Ensures a quick response and resolution of issues/risks (A1, B3):*

Both case studies agreed that the hybrid APMM (ver. 0) ensures a quick response and resolutions to project risks and issues.

- *Ensures minimised and simplified documentation/processes/semantics and focuses on completing the work (A1, A2, A3, B3):*

Both case studies were in agreement that the hybrid APMM (ver. 0) Ensures minimised and simplified documentation/processes/semantics to allow team members to focus more on completing the work.

- *Continuous improvement takes place (A2, B1, B2, B3):*

Both organisations agree that the hybrid APMM (ver. 0) allows for continuous improvement of deliverables to take place to ensure that products of quality are delivered.

- *Is an integrated approach, enhances integration and ensures integrated communication (highlight: collaborative development phase; A1, A3, B1, B2):*

Both of the case studies were in agreement that the hybrid APMM (ver. 0) is an integrated approach that enhances integration and ensures transparent and integrated communication.

- *Is a simple process (A1, A2, A3, B2, B3):*

Both case studies strongly agreed that the hybrid APMM (ver. 0) is a simple process.

- *Is acceptable, implementable, flexible and adaptable to its surrounding environment (A1, A2, A3, B1, B3):*

Both organisations strongly agree that the hybrid APMM (ver. 0) is acceptable to practitioners, implementable, and flexible with the ability to adapt itself to its surrounding project and business environment.

- *Is a new and fresh approach (A1, B3):*

Both case studies agreed that the hybrid APMM (ver. 0) is a new and fresh approach to project management.

- *Is not a traditional approach and has the possibility of performing better than traditional approaches (A1, A2, A3, B1, B3):*

Both organisations strongly supported the advantage that the hybrid APMM (ver. 0) is not a traditional approach and has the possibility and potential ability to perform better than other traditional approaches such as PMBOK and PRINCE2.

- *Keeps client/stakeholder constantly involved and satisfied (A1, A2, A3, B1, B2, B3):*

All the participants in both organisations strongly supported that the hybrid APMM (ver. 0) ensures constant customer/ stakeholder involvement and satisfaction.

- *Provides a suitable change-request form to manage change (A1, A2, A3, B1, B2, B3):*

Both the case studies strongly agreed that the change-request form that is provided by the hybrid APMM (ver. 0) is suitable for managing change requirements.

- *Will assist project managers and teams to deliver projects successfully in constantly changing project environments (A1, A2, A3, B2, B3):*

Both case studies strongly agreed that the hybrid APMM (ver. 0) assists the whole team to deliver projects successfully in a constantly changing project and business environment.

The advantages of the hybrid APMM (ver. 0) that were supported only by the large organisation are:

- *Ensures that the responsibilities of a project manager are not taken away (A1):*
The senior manager of the large organisation was the only participant that mentioned the advantage that the hybrid APMM (ver. 0) ensures that a project manager's responsibilities are not taken away because the hybrid APMM (ver. 0) will only assist the project manager in managing the project. It is obvious that the hybrid APMM (ver. 0) cannot be solely responsible for the success of a project.
- *Enforces a culture in which people can embrace change (A1):*
According to the senior manager of the large organisation, the hybrid APMM (ver. 0) enforces a culture in which people must embrace change in order to remain competitive in an environment that is ever changing.
- *Ensures that benefit realisation is completed (A1):*
The senior manager of the large organisation stated that the hybrid APMM (ver. 0) ensures that regular benefit realisation is done to ensure that what was promised is delivered when the project reaches completion.
- *Monitors and controls deliverables (A1):*
The senior manager of the large organisation commented that the hybrid APMM (ver. 0) ensures that deliverables are monitored and controlled regularly to ensure that the project is well controlled.
- *Will deliver project within time, cost, and quality constraints (A3):*
The team member of the large organisation explained that the hybrid APMM (ver. 0) will have the ability to deliver a project within time, cost and quality constraints.
- *Allows speculation on the unknowns and planning on the facts (A1):*
According to the senior manager of the large organisation, the hybrid APMM (ver. 0) allows speculation on uncertainties, while the team can plan based on the known facts.
- *Allows the team to achieve its objectives more quickly (A2, A3):*
The project manager and team member of the large organisation agreed that the hybrid APMM (ver. 0) allows a team to achieve the project objectives more quickly than normal.

The advantages of the hybrid APMM (ver. 0) that were supported only by the small organisation are:

- *Ensures that lessons learnt are implemented (B2):*
The project manager of the small organisation stated that the hybrid APMM (ver. 0) ensures that lessons learnt are implemented as the project progresses.
- *Focuses on incremental and iterative development (B1, B2, B3):*
All the participants of the small organisation explained the advantage of the hybrid APMM

(ver. 0) that focuses on planning and developing projects incrementally, where each increment is completed iteratively.

- *Assists in managing uncertainties (B2, B3):*

The project manager and team member agreed that the hybrid APMM (ver. 0) assists the team in managing uncertainties more effectively.

- *Provides clear direction and vision (B2):*

The project manager of the small organisation was the only participant who explained that the hybrid APMM (ver. 0) has the advantage of providing clear direction and a unified vision.

Theme 6: Shortcomings of the hybrid agile project management methodology (ver. 0)

All the shortcomings and disadvantages identified from the interviews relating to the hybrid APMM (ver. 0) were addressed by identifying a workable solution for each shortcoming and incorporating it into the hybrid APMM (ver. 1). There were no shortcomings of the hybrid APMM (ver. 0) that were supported by both the large and small organisations.

The shortcomings of the hybrid APMM (ver. 0) that were supported only by the large organisation are:

- *Needs to include a stakeholder mapping (A1):*

Although the senior manager of the large organisation stated initially that she would not add anything to the hybrid APMM (ver. 0), later in the interview, she explained that she would include a stakeholder mapping, to ensure that all the stakeholders are identified and that their reporting structures are understood. This is an important aspect, which was addressed by the hybrid APMM (ver. 1) because in most projects stakeholders are the most important people to keep satisfied.

- *Requires a project manager who enforces and manages the process effectively (A1):*

The senior manager mentioned that the hybrid APMM (ver. 0) needs a project manager who enforces and manages the hybrid APMM (ver. 0) effectively. Although this is a people factor and not a direct disadvantage of the hybrid APMM (ver. 0) itself, it will be addressed by the hybrid APMM (ver. 1) to ensure that a skilful and strong project manager is hired to execute the PMM.

The shortcomings of the hybrid APMM (ver. 0) that were supported only by the small organisation included the following:

- *Must be careful not to discuss the same thing repeatedly (B3):*

The team member of the small organisation was the only participant who offered a word of caution that when the hybrid APMM (ver. 0) is executed that the same issues are not

discussed repeatedly, especially in the improve and adapt step of the iterative refactoring cycle. This concern was addressed by the hybrid APMM (ver. 1).

- *Must not give team members free rein – need critical decision-makers (B1):*

The senior manager of the small organisation emphasised that the team members must not feel that they have free rein and do what they want when they want because it is easy to log a small change request. This was addressed by the hybrid APMM (ver. 1), where change request must be minimised; otherwise, it becomes an administrative burden to log all the changes that occur. There must also be strict control on the deliverables and the project has to have critical decision-makers – people who assess the risks and make decisions, instead of only discussing the problem.

- *Regular sign-offs are counterproductive (B1, B3):*

The senior manager and team member of the large organisation emphasised that regular sign-offs are counterproductive, as people waste time trying to obtain sign-offs, while some aspects of the project stand still. This was addressed by the hybrid APMM (ver. 1).

6.2.3 Discussion of cross-case analysis results and suggested improvements to the hybrid agile project management methodology (ver. 0)

The findings and recommendations after the completion of the cross-case analysis were used to enhance the hybrid APMM (ver. 0) before distributing it to respondents with a questionnaire for further evaluation and improvement. The hybrid APMM (ver. 0) was adjusted and improved to ensure that it addressed and incorporated the propositions of themes 1, 2, 3, 4 and 6. In this section, each proposition will be examined in order to determine whether the hybrid APMM (ver. 0) already addressed it. If it was not addressed, a solution was identified and the manner in which the hybrid APMM (ver. 0) was improved will be explained in this section. The improvements made to the hybrid APMM (ver. 0) resulted in the hybrid APMM (ver. 1).

Theme 1: Critical project success factors

- *Consider the nature of a project before applying a PMM/ASDM (A2, A3, B2, B3):*

In order to incorporate this proposition, an extra objective was added to the *pre-initiation phase* of the hybrid APMM (ver. 0), where the unique characteristics of the project will be identified. The unique characteristics will be identified by estimating how well each of the twenty eight APSFs of the *agile project success criteria* are addressed by the project. For example, to estimate whether the *management commitment* APSF is addressed by the project, the following question should be asked; is management committed to the project? If 'yes', its addressed, if 'no'; What can be done to get management committed to the project?

These types of questions will be asked for each APSF to determine the characteristics, complexity and uniqueness of the project. The project/system critically will also be defined using the framework of Crystal ASDMs (see Figure 2.11), which would contribute as a unique characteristic of the project. These characteristics will be used to determine whether the hybrid APMM (ver. 1) is the correct approach to follow for the project. If the hybrid APMM (ver. 1) is chosen as the PMM to execute the project, the unique characteristics of the project will be reviewed throughout the duration of the project by adding it as an activity that must be completed as part of the *adapt, direct, monitor and control* phase as well as the *direct, monitor and control* step of the *iterative refactoring cycle*. This will ensure that the project keeps the changing nature of the project in mind and adapts to it accordingly.

- *Do benefit realisation regularly (A1, A2, A3, B1, B2, B3):*
This is addressed by the hybrid APMM (ver. 0), where the benefit realisation is completed as part of the *iterative refactoring cycle* and at a *collaborative increment* level. It was also identified as an advantage of using the hybrid APMM (ver. 0) by the senior manager of the large organisation (theme 5: *ensures that benefit realisation is completed [A1]*).
- *Do maintenance after implementation (A1, A2, A3, B1, B2, B3):*
This is addressed by the hybrid APMM (ver. 0) *post-project maintenance and continual improvement* phase.
- *Formally close-out, sign-off and hand-over a project (A1, B2, B3):*
This is addressed by the hybrid APMM (ver. 0) in the *close-out and hand-over* phase, as well as the regular sign-offs during the collaborative increments.
- *Integrated communication (A1, A3, B1, B2):*
Integrated communication is integrated throughout the hybrid APMM (ver. 0). Both the large and small organisations explained that this is one of the advantages of utilising the hybrid APMM (ver. 0) because it *is an integrated approach that enhances integration and ensures integrated communication (A1, A3, B1, B2)*.
- *Keep PMM/ASDM simple, minimise documentation (keep important documents) and focus on completing the work (A1, A2, A3, B1, B2, B3):*
This proposition is addressed by the hybrid APMM (ver. 0), where the participants of both case studies supported the advantages of it being simple (*is a simple process [A1, A2, A3, B2, B3]*), minimising documentation and focuses on completing the work (theme 5: *Ensures minimised and simplified documentation/processes/semantics and focuses on completing the work [A1, A2, A3, B3]*).
- *Keep stakeholders involved and provide regular feedback – this will ensure easier sign-offs (A1, A2, A3, B1, B2):*
This proposition is addressed by the hybrid APMM (ver. 0), where the client and stakeholders are constantly involved as the project progresses. Both organisations strongly

support the advantage the hybrid APMM (ver. 0) has of constantly involving stakeholders because it will result in easier and quicker sign-offs (theme 5: *keeps client/stakeholder constantly involved and satisfied* [A1, A2, A3, B1, B2, B3]).

- *Log and implement lessons learnt as the project progresses to prevent repeating mistakes* (A1, A2, A3, B1, B2, B3):

This is addressed by the hybrid APMM (ver. 0) in the *learn* step of the *iterative refactoring cycle*, to guarantee that all lessons are identified and logged on a regular basis to ensure that the same mistakes are not made more than once. This proposition was also supported as an advantage of the hybrid APMM (ver. 0) by the project manager of the small organisation (theme 5: *ensures that lessons learnt are implemented* [B2]).

- *Manage changing/unclear requirements – give stakeholders what they want, not what developers think they need* (A2, B1, B2, B3):

One of the fundamental aspects of the hybrid APMM (ver. 0) is managing and adapting to changing requirements and a changing environment on a regular basis. This proposition is addressed by the *adaptive actions* in the *collaborative develop* phase and *interactive refactoring cycle*. The project team can manage and adapt to changes, in the *improve and adapt* step and the *direct, monitor and control* step of the *interactive refactoring cycle*, and in the *adapt, direct, monitor and control* phase. Changing requirements are also addressed and managed by the change-request form. Both the large and small organisations viewed this proposition as an advantage of the hybrid APMM (ver. 0; theme 5: *is acceptable, implementable, flexible and adaptable to its surrounding environment* [A1, A2, A3, B1, B3], *will assist project managers and teams to deliver projects successfully in constantly changing project environments* [A1, A2, A3, B2, B3]). The hybrid APMM (ver. 0) ensures that stakeholders' requirements are satisfied through regular benefit realisation and keeping the stakeholders involved as the project progresses.

- *Need a strong project manager/leadership* (A2, A3, B1, B2, B3):

The only way to address this proposition is to ensure that a strong project manager with the correct skills and expertise is appointed to manage the project by applying the PMM/ASDM. This was added to as a key objective of the *pre-initiation* phase in the hybrid APMM (ver. 1).

- *Need a PMM/SDM that is flexible and agile – with the ability to adapt to change* (A1, A2, A3, B1, B3):

The hybrid APMM (ver. 0) was designed with main focus of being flexible with the ability to adapt to change. As with the proposition mentioned earlier, both the large and small organisations viewed this proposition as an advantage of the hybrid APMM (ver. 0; theme 5: *is acceptable, implementable, flexible and adaptable to its surrounding environment* [A1, A2, A3, B1, B3], *will assist project managers and teams to deliver projects successfully in constantly changing project environments* (A1, A2, A3, B2, B3)

- *Need integrated project management and system development (A1, A3, B2):*
The hybrid APMM (ver. 0) was designed to integrate and unite all the areas and aspects of project management and system development. Both the large and small organisations viewed this proposition as an advantage of the hybrid APMM (ver. 0; *is an integrated approach, enhances integration and ensures integrated communication [A1, A3, B1, B2]*).
- *Need the right people/resources on a team that is not short staffed with the right skills, allocated correctly and working together with the same vision – will result in a higher probability of project success (A1, A2, A3, B1, B2, B3):*
The only way to address this proposition is to ensure that people with the required skills and expertise are appointed to be part of the project team that will execute the project. This was also added as one of the key objectives together with the proposition above (ensure that a strong project manager is assigned) to be executed in the *pre-initiation* phase of the hybrid APMM (ver. 1).
- *Regular sign-offs are important, especially in a changing environment (A1, A2, A3, B2):*
This is addressed by the hybrid APMM (ver. 0), where the regular sign-offs happen after every collaborative increments has been completed. An important word of caution will be added to the hybrid APMM (ver. 0): regular sign-offs can be counterproductive if not managed correctly. This will be addressed in theme 6 below.
- *Require proper monitoring and control procedure (A1, A2):*
The *direct, monitor and control* step in the *iterative refactoring cycle* and the *adapt, direct, monitor and control* phase in the hybrid APMM (ver. 0) addresses this proposition. Furthermore, the proposition is supported as an advantage of the hybrid APMM (*monitors and controls deliverables [A1]*).
- *Stakeholder mapping (A1):*
This factor will be addressed in theme 6 to address the hybrid APMM (ver. 0).
- *Undefined/unclear scope (at beginning) – results in a great deal of rework (A1, A2):*
This factor is addressed by the hybrid APMM (ver. 0) in the *preparation* phase, where as many user and business requirements as possible are gathered beforehand to decrease the change for new and changing requirements. This will result in much less rework, as most main requirements will already have been gathered before the project start execution.
- *Deliver quickly (A1):*
This is addressed by the hybrid APMM (ver. 0), where functionality is regularly released in the iterative refactoring cycle. The main aim of focusing on incremental and iterative development is to deliver functionality quickly to the business for them to use it and provide possible improvements required.
- *Inform people of changes and the project managers must ensure that the changes are enforced (A1):*

An agenda item was added to the *post-incremental meeting* and during the *daily stand-up meetings* in the hybrid APMM (ver. 0) to ensure that everyone knows of the changes that occur in the project. An activity to provide a regular organisation communiqué was also added as an activity in the *adapt, direct, monitor and control* phase to ensure that the business and the project are up to date on the progress of and possible changes to the project that might affect the employees of the organisation. One advantage of the hybrid APMM (ver. 0) as explained by the senior manager of the large organisation is that it *enforces a culture in which people can embrace change (A1)*.

- *Ensure people are accountable for their tasks (A2):*

The hybrid APMM (ver. 0) ensures regular interaction amongst all the individuals involved in the project but it cannot ensure that people fulfil their responsibilities. This can only be addressed if people who are diligent and hardworking are appointed together with a strong project manager who can lead and drive the project to successful completion. This further emphasises the addition of a key objective in the *pre-initiation phase* by ensuring that the correct team members and project manager(s) with the right skills are appointed to execute the project.

- *Know your stakeholders and team (A3):*

The best way to get to know the team members and stakeholders is to have regular interaction with them. This proposition is addressed in same way, where the project manager can get to know his/her team members better during the *daily stand-up* meetings and daily interaction. In order to come to know people, their backgrounds must be understood and most people will only open up to each other if there is a level of trust, which can be created by having social and team-building events. The hybrid APMM (ver. 0) will be improved by having a celebration session when an increment is completed directly after the post-incremental meeting, where people can connect at a social level. If funding is available team-building sessions/golf day or match days (watch a game of rugby or soccer together) can be organised, where all the people involved in the project can connect socially.

- *Need adequate funding/budget (A1):*

This funding was addressed by adding a key objective in the *pre-initiation* phase, where assurance from the stakeholders must be received, preferably as a signed document stating that the organisation has the necessary funding to commence and complete the project after taking contingencies, possible project changes, and project risks into consideration.

- *Project manager and team members must be on board to utilise and enforce a PMM (B1):*

The hybrid APMM (ver. 0) cannot ensure that the project manager follows it step by step. The organisation must either make a PMM the standard for managing projects in their organisation or the project manager and team members must be confident that a PMM will

work. Their confidence will be supported by evidence that the selected PMM was successful in a similar project and business environment.

Theme 2: Environmental characteristics

- *Constantly fast-changing environment (dynamic) – change is the only constant (A1, A2, A3, B1, B2, B3):*

The hybrid APMM (ver. 0) was designed to be agile with the ability to adapt to a constant and fast-changing project environment. As with theme 1, both the large and small organisations viewed this proposition as an advantage of the hybrid APMM (ver. 0) because it has the ability to deliver projects where change is the only constant (theme 5: *is acceptable, implementable, flexible and adaptable to its surrounding environment [A1, A2, A3, B1, B3], will assist project managers and teams in delivering projects successfully in constantly changing project environments [A1, A2, A3, B2, B3]*).

- *Stakeholders expect things to change immediately and they expect/force team members to deliver to unrealistic deadlines (A1, A2, A3, B1, B3):*

The hybrid APMM (ver. 0) cannot ensure that project deliverables are delivered to unrealistic deadlines. One way to address this proposition in the hybrid APMM (ver. 0) is to ensure that a skilful project manager is assigned to the project in the pre-initiation phase, who can negotiate with stakeholders to ensure that they have realistic expectations that can be met within the time, scope, quality and financial constraints provided.

- *Project environment is challenging (A1):*

Although the hybrid APMM (ver. 0) has the ability to deliver projects in a challenging environment because of its ability to adapt, this proposition was addressed further by determining a project's nature in the *pre-initiation* phase of the hybrid APMM (ver. 1) before a PMM is applied, as every project environment is unique and challenging.

- *Team members feel that the project manager does not always understand (B3):*

The main reason that project managers do not understand is that there is no transparent and integrated communication taking place. If a team member experiences a problem of any nature there must be an open communication channel to discuss the problem/issue with the project manager in order to find a solution to the problem.

- *Users do not know what they want (A1):*

If users cannot make up their minds of what exactly they need, the project manager must make it clear to them that the project will be jeopardised by their uncertainty. If the matter cannot be resolved in this manner, the project manager must escalate it to management to intervene to ensure that the project is delivered within time, cost, scope and quality constraints.

Theme 3: People factors

- *Human, behavioural, political and cultural factors are difficult to manage (A1, A3, B1):*

This proposition cannot be addressed fully by any PMM. The best way to manage different cultures, behaviours and religions is to treat people with respect (Holcombe & Holcombe, 2008:25). If everyone agrees to treat one another with respect and to focus on completing the work, this will result in fewer issues related to this proposition. Treating everyone involved in the project with respect at all levels is re-emphasised in the *adapt, direct, monitor and control* phase of the hybrid APMM (ver. 1).

- *Performance management – people must be rewarded for doing a good job (A3, B1):*

This is a very important proposition, which must be implemented in any organisation and project, because it is the workforce that ensures that the project is completed. People feel that they have purpose if they have been rewarded for doing good work.

- *People talk too much and do too little (B3):*

This proposition cannot be addressed by any PMM. It is the project manager's responsibility to ensure that discussions are related to the project and the completion of deliverables. Social discussions have to happen to ensure that the team members and stakeholders become acquainted with one another, but if it is going to jeopardise the project deliverables the project manager must refocus all the people involved to ensure that the deliverables are completed in time.

The general CPSF propositions of theme 1, and the propositions of themes 2 (environmental characteristics) and 3 (people factors) identified through content and cross-case analysis are grouped and compared with the existing *agile project success criteria* (see Table 5.4) to identify any new APSFs. The reason for grouping the propositions of themes 2 and 3 under the *agile project success criteria* are because they are just like theme 1, CPSFs (environmental and people factors) required for the successful completion of a project. In essence, all projects have some agile elements because most of them are executed in a constantly changing environment, which is the reason that the propositions of themes 1, 2 and 3 were grouped under the different APSFs of the *agile project success criteria* using *blue* text (see Table 6.1). If a proposition could not be grouped under any existing APSF, it was identified as a new APSF which was presented in *green* text (see Table 6.1). The result of this comparison and grouping is presented in Table 6.1.

The propositions of theme 4 were not grouped under the existing *agile project success criteria* of Table 6.1, because these propositions are related to *project management and system development principles* - they are not CPSFs required for a project to be completed successfully. If the propositions of theme 4 must be grouped under the *agile project success*

criteria, all the propositions of theme 4 would fall under the *project management process* APSF. Propositions of themes 5 and 6 were also not grouped under the *agile project success criteria* in Table 6.1, because they are also not CPSFs required for the successful completion of a project. The propositions of themes 5 and 6 represent the advantages and shortcomings of the hybrid APMM (ver. 0).

Table 6.1: Supporting CPSF grouping

Agile project success criteria	Supporting CPSFs and propositions
1. Management commitment	Executive/senior management ownership and support.
2. Organisational environment	- Adequate organisational change management. - Stakeholders, users and team members must understand the benefits of successful project delivery for the business. - Alignment of project and organisational objectives/vision. - Impact analysis of the project on the organisation. - Considering personal interests and ambition, but not placing these before the organisation's interests. - Understanding existing business processes. - Clearly defined and understood business drivers and objectives. - Attempt to re-engineer business.
3. Team environment	Effective, aligned and cross-functional team organisational structure.
4. NEW: Project selection and prioritisation	Projects must be selected based on an accurate and sound business plan.
5. Team capability	- Trained, experienced and skilled project managers and team members. - Good leadership. <i>Need a strong project manager/leadership (A2, A3, B1, B2, B3).</i> <i>Need the right people/resources on a team that is not short staffed with the right skills, allocated correctly and working together with the same vision – will result in a higher probability of project success (A1, A2, A3, B1, B2, B3).</i>
6. Client involvement	- User/client/stakeholder/sponsor involvement, participation, commitment, support, accountability and ownership. - Consultation with stakeholders/users/clients. - Considering stakeholders' employees' recommendations, ideas and solutions. <i>Keep stakeholders involved and provide regular feedback – this will ensure easier sign-offs (A1, A2, A3, B1, B2).</i>
7. NEW: Role definition	- Clearly defined roles, authority and responsibilities. - Understanding executive/senior management role in the project. <i>Stakeholder mapping (A1).</i>
8. NEW: Communication	- Constant and effective communication and feedback. - Willingness to share information. - Willingness to seek the advice of senior management. <i>Integrated communication (A1, A3, B1, B2).</i>
9. NEW: Client expectations	- Managing unrealistic client and business expectations. - Satisfied client, user, business and stakeholder expectations and requirements. - Clearly defined, gathered and controlled user and business requirements and expectations. <i>Stakeholders expect things to change immediately and they expect/force team members to deliver to unrealistic deadlines (A1, A2, A3, B1, B3).</i> <i>Users do not know what they want (A1).</i>
10. NEW: Politics and culture	- Managing politics and culture <i>Human, behavioural, political and cultural factors are difficult to manage (A1, A3, B1).</i>
11. NEW: Collaborative teamwork	- A project requires a collaborative effort. - Wellness and safety of team members are considered to be of high importance. - Motivated team members. - Quick and fair resolution of conflict. - Team-member commitment. - Self-controlled project team. <i>Know your stakeholders and team (A3).</i> <i>Project manager and team members must be on board to utilise and enforce a PMM (B1)</i> <i>Team members feel that the project manager does not always understand (B3)</i> <i>Performance management – people must be rewarded for doing a good job (A3, B1)</i> <i>People talk too much and do too little (B3).</i>
12. Project management process	- Adherence to high-quality standards and constraints for deliverables and the project. - Defined checkpoints, gate reviews and structure. - Correct application of project management theory. - Correct selection of the correct PMM. - Use and acceptance of the benefit of a defined, effective and standardised methodology. - Understanding the advantage of project management. - Processes and policies are defined with regard to project management best practices,

	approaches and concepts. - Knowing that project management does not happen at executive level. - Preventing scope creep - Reliable estimates. - Adequate cash flow management and fund provision. - Suitable and timely sign-offs. - Managing external project factors, such as vendor and regulatory issues. <i>Need adequate funding/budget (A1).</i> <i>Do benefit realisation regularly (A1, A2, A3, B1, B2, B3).</i> <i>Keep PMM/ASDM simple, minimise documentation (keep important documents) and focus on completing the work (A1, A2, A3, B1, B2, B3).</i> <i>Need a PMM/SDM that is flexible and agile – with the ability to adapt to change (A1, A2, A3, B1, B3).</i> <i>Need integrated project management and system development (A1, A3, B2).</i>
13. Project definition process	- Clearly defined, understood and controlled business case, benefits, objectives, deliverables and project definition. - Clear and unified vision/focus. - Defining and minimising a robust project scope and controlling scope changes. <i>Undefined/unclear scope (at beginning) – results in a great deal of rework (A1, A2).</i>
14. NEW: Change management and control	- Effective change identification, control, management, processes, and procedures. <i>Manage changing/unclear requirements – give stakeholders what they want, not what developers think they need (A2, B1, B2, B3).</i> <i>Inform people of changes and the project managers must ensure that the changes are enforced (A1).</i>
15. NEW: Monitoring and control	- Good quality control, management, processes, procedures, and acceptance criteria. - Understandable monitoring and controlling processes and procedures to track project progress. - Controlled project delivery. - Standardised reporting and monitoring procedures. - Resolving project problems identified as soon as possible. - Maintaining and updating the WBS. - Maintaining the project approval and sign-off processes. <i>Require proper monitoring and control procedure (A1, A2).</i>
16. NEW: Reporting and tracking	- Tracking and adhering to cost and budget estimates. - Standardised reporting and monitoring procedures. - Gathering lessons learnt as the project progresses. <i>Log and implement lessons learnt as the project progresses to prevent repeating mistakes (A1, A2, A3, B1, B2, B3).</i>
17. NEW: Risk and issue management	Regularly and promptly identifying, assessing, mitigating, resolving, reviewing, monitoring and controlling project risks and issues.
18. Agile software techniques	
19. Delivery strategy	- Use of prototyping. <i>Do maintenance after implementation (B1, B2, B3).</i> <i>Deliver quickly (A1).</i> <i>Formally close-out, sign-off and hand-over a project (A1, B2, B3).</i>
20. NEW: Technical designs, practices and tools	- Sufficient technical design, practices and utilisation of planning tools. - Performing a return on investment assessment. - No overreliance on technology. - Appropriate use and evaluation of, and experimentation with technology. - Standard software infrastructure.
21. NEW: Testing	High-quality testing.
22. Project nature	- Avoiding direction change in the middle of the project. - Considering the growth and evolution of a project. <i>Consider the nature of a project before applying a PMM/ASDM (A2, A3, B2, B3).</i>
23. Project type	
24. Project schedule	- Developing, tracking, adhering to, and correctly executing a detailed/robust project schedule and plan (verbal advice). - Acknowledgement that the project schedule and cost are inseparable. - Adequate budgets and realistic timelines.
25. NEW: Resource management (availability and allocation)	- Availability of resources and technical competencies (skills, hardware, software, sufficient funds, etc.). - Sufficient resources (money, time, expertise, etc.). - Proactive resource allocation. - Correct team and project manager selection and organisation. <i>Need the right people/resources on a team that is not short staffed with the right skills, allocated correctly and working together with the same vision – will result in a higher probability of project success (A1, A2, A3, B1, B2, B3).</i> <i>Ensure people are accountable for their tasks (A2).</i>
26. NEW: Project size	Sufficient project sizing.
27. NEW: Project strategy	- Appropriate project strategy. <i>Regular sign-offs are important, especially in a changing environment (A1, A2, A3, B2).</i>
28. NEWEST: Project environment	<i>Constantly fast-changing environment (dynamic) – change is the only constant (A1, A2, A3, B1, B2, B3).</i> <i>Project environment is challenging (A1).</i>

From the twelve APSFs of Chow and Chao (2008:964), only six (delivery strategy, agile software-engineering techniques, team capability, project management process, team

environment, client involvement) were found by Chow and Chao (2008:964) to be relevant to agile projects, while there was no evidence for the other six APSFs as being prerequisites for the success of agile projects. After the CPSFs (identified from the completed case studies) were grouped under the twenty-seven APSFs in Table 6.1, it was found that they supported six of the twelve APSFs of Chow and Chao (2008:964). Four (delivery strategy, team capability, project management process, client involvement) of the six APSFs of Chow and Chao (2008:964) that are relevant to agile projects were supported by the CPSFs identified in the case studies.

After comparing the propositions of the case studies with the APSFs of Table 5.4, two propositions from theme 2 (*constantly fast-changing environment (dynamic) – change is the only constant (A1, A2, A3, B1, B2, B3)* and *project environment is challenging (A1)*) in the case studies could not be grouped under the twenty-seven main APSFs. This resulted in a newly identified APSF called *project environment*, which was added to the *agile project success criteria*.

Theme 4: Project management and system development principles

- *Incremental and iterative planning and development (A1, A2, A3, B1, B2, B3):*
The hybrid APMM (ver. 0) addresses this proposition because it was designed to develop and deliver a project incrementally with the *collaborative increments*, where every increment is completed iteratively with the *iterative refactoring* cycle. This factor was supported as an advantage of the hybrid APMM (ver. 0) by some of the participants of the large and small organisations (theme 5: *focuses on incremental and iterative development [B1, B2, B3]*).
- *Rather speculate than plan because planning only eliminates some uncertainty – not all of it (A1, A2, A3, B1, B2, B3):*
The APMM (ver. 0) addresses this proposition in the *preparation* phase and in the *speculate and analyse* step in the *iterative refactoring cycle* where it focuses on speculating when uncertainties are present, while planning on certain facts. This factor was supported as an advantage of the hybrid APMM (ver. 0) by the senior manager of the large organisation (theme 5: *Allows speculation on the unknowns and planning on the facts [A1]*).
- *Continuously review and improve the deliverables (A2, B2):*
This proposition is addressed by the hybrid APMM (ver. 0) by executing the *test and evaluate* and *improve and adapt* steps in the *iterative refactoring cycle* to ensure that the team continually reviews and improves deliverables or sub-deliverables. This factor was supported by some of the participants of the large and small organisations as an advantage of the hybrid APMM (ver. 0; theme 5: *continuous improvement takes place [A2, B1, B2,*

B3]).

- *Practitioners do apply project management and system development principles (A2, A3, B1, B2, B3):*

Because practitioners do apply some project management and system development principles it will result in a higher probability of the hybrid APMM (ver. 0) being accepted in practice because the hybrid APMM (ver. 0) contains the best practices of both ASDMs and PMMs.

- *Practitioners with knowledge of PMBOK/PRINCE2 do not apply it theoretically/fully by the book, only partially, because practically as a whole they do not work (A1, A2, A3, B1, B2, B3):*

The hybrid APMM (ver. 0) cannot support this proposition because practically as whole it has not been tested in practice. The hybrid APMM (ver. 2) will still contain flaws and it will also not be able to deliver all types of projects because there is no “best breed” PMM or ASDM, but the hybrid APMM (ver. 0) does have the advantage of having the ability to adapt to projects of different types and sizes. This advantage was supported by all the participants of the large and small organisations (theme 5: *adjusts to different types and different size projects* [A1, A2, A3, B1, B2, B3]).

- *Practitioners do not follow formal change-management procedures to manage change (A3):*

This proposition or concern can be addressed by the hybrid APMM (ver. 0), as it follows effective change management in the *adapt, direct, monitor and control* phase, the *direct, monitor and control* step in the *iterative refactoring cycle*, the *adaptive actions* in the *collaborative development* phase and by the completion of a *change-request form*. All the participants of the large and small organisations viewed the use of the change-request form to manage change as an advantage of the hybrid APMM (ver. 0; theme 5: *provides a suitable change-request form to manage change* [A1, A2, A3, B1, B2, B3]).

- *Feedback meetings have to be purpose driven and the right people must be invited to these meetings (B2):*

This proposition will be addressed by adding key outcomes and key decisions that must be made to the agenda of a meeting to ensure that the meeting has a specific purpose (stand-up, pre-incremental meeting, post-incremental meeting, feedback meetings, etc.). It will be the project manager’s responsibility to ensure that the correct people are invited to a meeting and it will be the chairperson’s responsibility to ensure that the discussions are focused on project-related topics.

- *There is no “best breed” PMM/ASDM (A1, A2, A3, B1, B2, B3):*

As explained in the proposition above, the hybrid APMM (ver. 0) is not trying to be a “best breed” PMM or ASDM, but one that can adapt to change, and projects of different sizes and types.

Theme 5: Advantages of the hybrid agile project management methodology (ver. 0)

The propositions related to theme 5 were used to support the propositions in theme 1 (CPSFs) and theme 4 (*project management and system development principles*). Theme 5's propositions will not be discussed further in this section because these propositions were already addressed by the hybrid APMM (ver. 0) and they were viewed by the participants as advantages of the hybrid APMM (ver. 0).

Theme 6: Shortcomings of the hybrid agile project management methodology (ver. 0)

There were some shortcomings identified, that were addressed to improve the hybrid APMM (ver. 0).

- *Must be careful not to discuss the same thing repeatedly (B3):*

In order to ensure that the same topics are not discussed repeatedly, meetings must be purpose driven as explained in theme 4 – *feedback meetings have to be purpose driven and the right people must be invited to these meetings (B2)*. The suggested solution in theme 3 for *people talk too much and do too little (B3)* was also applied to address this proposition.

- *Must not give team members free rein – need critical decision-makers (B1):*

This is an important statement because team members must still have the understanding that a *change request* or *adaptive action* may only be done if it is absolutely necessary. This will be addressed by the hybrid APMM (ver. 0) by adding the responsibility to that of the project manager, where he/she must keep steady control of the project team to ensure that they deliver on timelines promised. Only unforeseen circumstances or changes that have a direct impact on the success of a project may be used as excuses to log a *change request* or to perform an *adaptive action*.

- *Regular sign-offs are counterproductive (B1, B3):*

Regular sign-offs are counterproductive if not managed correctly. This will be addressed by the hybrid APMM (ver. 0) by only having incremental sign-offs when a *collaborative increment* has been completed for large projects. If a small to medium-sized project's *collaborative increment* has been completed, a sign-off is not required to continue with the following collaborative increments. The advice however will still be provided that regular sign-offs must be done where a set of collaborative increments can be combined (depending on the size of the project) into one sign-off to ensure that the client does not later claim that a set of deliverables was not what they required. If the project is very small, only one sign-off is required when the project is completed. It would also be clearly stipulated in the *collaborative development* phase that team members may not wait for sign-offs to continue with their work. Whether it is a large or small project, the next *collaborative*

increment can be commenced without a sign-off of previous dependable increments. If there are changes, they can always be addressed with the change-control procedures embedded throughout the hybrid APMM (ver. 0).

- *Needs to include a stakeholder mapping (A1):*

The hybrid APMM (ver. 0) will address this proposition by including the set up of a stakeholder map as one of the key objectives that must be achieved in the *vision and definition* phase. The stakeholder map will also be added as an activity in the adapt, direct, monitor and control phase, to ensure that the stakeholder map is kept up to date regarding any changes that might have occurred. The stakeholder map will furthermore ensure that all stakeholders are kept updated on the progress of the project.

- *Requires a project manager who enforces and manages the process effectively (A1):*

This proposition was addressed in theme 1 – *project manager and team members must be on board to utilise and enforce a PMM (B1), need a strong project manager/leadership (A2, A3, B1, B2, B3)*.

The changes and improvements made to the hybrid APMM (ver. 0) after the case studies were completed will be explained in Section 7.3, where the final hybrid APMM (ver. 2) will be presented. All the solutions of the propositions explained in Section 6.2.3 were incorporated into the hybrid APMM (ver. 0) to create the hybrid APMM (ver. 1), which was used in the survey as will be explained in Section 6.3.

6.3 Evaluation of the hybrid agile project management methodology (ver. 1)

After the hybrid APMM (ver. 1) had been developed, a survey was conducted as explained in Chapter 4 and was distributed to the various respondents with a questionnaire. The questionnaire contained the updated *agile project success criteria* (see Table 6.1) that were developed by combining the newly identified APSF (project environment) with the existing *agile project success criteria* (see Table 5.4). The questionnaire included the following questions:

- Question 1: *Background information (qualifications, position description, years of experience, organisation size).*

The organisation size could be selected from a dropdown list indicating whether the respondent work in a small (0–50 employees), medium-sized (50–100 employees) or large (>100 employees) organisation.

- Question 2: *Do you agree that the following agile project success factors are relevant to agile projects?*

The *agile project success criteria* and CPSF groupings in Table 6.1 were used and the respondent had to indicate whether the twenty-eight APSFs are relevant to agile projects on

a scale of 1 to 5, where:

- 1: disagree completely
 - 2: disagree
 - 3: neutral
 - 4: agree
 - 5: agree completely
- Question 3: *What other critical project success factors would you add for agile projects?*
 - Question 4: *Do you agree that the hybrid APMM (ver. 1) supports/addresses the following agile project success factors?*
- After presenting the hybrid APMM (ver. 1) to all the respondents a soft copy of the hybrid APMM (ver. 1) in PDF format was sent with the questionnaire for completion, as explained in Chapter 4. It was clearly stipulated in the e-mail containing the questionnaire and PDF that this question was only to be answered by the respondents if they had read and understood the hybrid APMM (ver. 1). The PDF could also be used as reference for the hybrid APMM (ver. 1) when required when completing the questionnaire. The *agile project success criteria* and CPSF groupings in Table 6.1 were used for this question, as with question 2. The respondent had to indicate whether the hybrid APMM (ver. 1) addresses the twenty-eight APSFs on a scale of 1 to 5 and answers were according to the same scale of 1 to 5.
- Question 5: *After reading the hybrid APMM document, what improvements do you suggest to the methodology and why?*

The purpose of including the supporting CPSFs in questions 2 and 4 of the questionnaire (see Annexure C) was not to evaluate each supporting CPSF individually, but to present what each evaluated APSF consist of.

6.3.1 Statistical analysis

The statistical analysis and results have been reviewed and approved by the Head of the Statistical Consultation Service, Dr. Suria Ellis at the Potchefstroom Campus of the North-West University.

The answers to the five questions asked were analysed and the findings were incorporated to improve the hybrid APMM (ver. 1). The statistical analysis of the answers will be explained in the following section.

6.3.1.1 Results of survey

The most important information required from question 1 was the organisation's size (large, medium and small), which was used to group the data being analysed. Of the questionnaires sent out to sixty-five respondents working in twenty-six different organisations in various industries (see Section 4.3.2.2) in and around Gauteng (South-Africa), fifty-one questionnaires (78.46%) was received. The questionnaires received from the respondents were from all twenty-six organisations and the various industries (see Section 4.3.2.2), where thirty-three were from large organisations, seven were from medium-sized organisations, and eleven were from small organisations.

The answers to questions 2, 3, 4, and 5 were statistically analysed in order to gain an understanding of the quantitative and qualitative data collected through the questionnaires. The answers to questions 2 and 4 were imported into Excel and STATISTICA in order to conduct a *one-way ANOVA* (Analysis Of Variance) and a *paired t-test*.

The purpose for conducting an ANOVA was to determine whether there was a difference in variance in the answers (rating) provided for the APSFs in the *agile project success criteria* between the large, medium-sized and small organisations.

The dependent t-test was used to assess whether the averages of questions 2 and 4 were statistically significant different from each other. If the average difference of the rating provided by respondents for each of the APSFs differed negatively, this would mean that the hybrid APMM (ver. 1) addresses the APSFs more than required or expected by the respondents, while if the difference is positive it would mean that the hybrid APMM (ver. 1) addresses the APSFs less than required or expected by the respondents (see Table 6.7).

T-tests and ANOVA tests were performed to test whether the average answer (rating) provided by the respondents in large, medium-sized and small organisations were statistically significantly different for a specific APSF. The results of the ANOVA and t-test are explained in more detail below.

The APSFs are prioritised for both questions 2 and 4. In question 2, the top ten APSFs regarding agile projects as rated by the fifty-one respondents are presented, while the top ten APSFs addressed by the hybrid APMM (ver. 1) as rated by the respondents are presented in question 4. The top ten APSFs in question 2 are compared with the top ten CPSFs in Table 5.1 of Kim *et al.* (2001:186) and the top ten CPSFs in Table 5.2 of Standish Group International (2001:4).

Question 2: Relevance of agile project success factors to agile projects (one-way ANOVA)

In question 2, respondents from large, medium-sized and small organisations indicated on a scale of 1 to 5 the extent to which they agreed that a specific APSF is relevant to an agile project. The maximum score for full agreement was thus a 5, and the score for total disagreement was a 1. The output of the *one-way ANOVA* provided the following results for the large, medium-sized and small organisations for each APSF.

Table 6.2: One-way ANOVA results for the relevance of APSFs to agile projects

Agile project success criteria	Large		Medium		Small		ANOVA	Welch
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std dev.	p-value	p-value
1. Management commitment	4.848	0.364	5.000	0.000	4.818	0.603	0.616	
2. Organisational environment	4.182	0.528	4.000	0.000	4.091	0.539	0.641	
3. Team environment	3.939	0.747	3.286	0.488	3.727	0.905	0.119	0.038
4. NEW: Project selection and prioritisation	3.939	0.933	3.857	0.900	3.545	1.036	0.498	0.564
5. Team capability	4.333	0.692	4.000	0.816	4.182	0.874	0.533	0.591
6. Client involvement	4.182	0.808	4.143	0.690	4.273	0.647	0.925	0.908
7. NEW: Role definition	4.152	0.870	4.143	1.069	4.000	1.000	0.893	0.908
8. NEW: Communication	4.394	0.747	3.857	1.345	4.273	1.009	0.365	0.602
9. NEW: Client expectations	3.606	1.029	3.857	0.690	3.909	0.944	0.612	0.596
10. NEW: Politics and culture	3.485	0.906	3.000	0.816	3.545	0.820	0.375	0.362
11. NEW: Collaborative teamwork	4.273	0.674	3.857	1.215	4.273	0.647	0.410	0.697
12. Project management process	4.061	0.659	3.714	0.756	4.091	0.831	0.475	0.542
13. Project definition process	4.121	0.696	3.714	0.756	4.273	0.647	0.248	0.316
14. NEW: Change management and control	3.970	0.810	4.000	0.816	4.091	0.944	0.918	0.933
15. NEW: Monitoring and control	4.152	0.870	4.286	0.756	4.182	1.079	0.938	0.921
16. NEW: Reporting and tracking	3.879	0.781	4.286	0.756	4.182	0.751	0.312	0.340
17. NEW: Risk and issue management	4.061	0.788	4.000	0.577	4.000	0.775	0.964	0.961
18. Agile software techniques	3.818	0.808	3.143	1.069	3.909	0.701	0.121	0.282
19. Delivery strategy	3.758	0.969	3.714	1.254	4.455	0.820	0.119	0.101
20. NEW: Technical designs, practices and tools	3.818	0.846	3.429	0.787	3.818	0.982	0.549	0.520
21. NEW: Testing	4.152	0.834	4.000	0.816	4.273	0.905	0.800	0.814
22. Project nature	3.727	0.876	3.857	0.690	4.182	0.874	0.319	0.368
23. Project type	3.545	1.034	3.429	1.134	3.364	1.286	0.883	0.903
24. Project schedule	3.970	0.883	3.571	0.787	4.000	1.000	0.541	0.498
25. NEW: Resource management	4.303	0.810	4.143	0.900	4.364	0.674	0.844	0.863
26. NEW: Project size	3.758	0.751	3.571	0.787	4.000	0.775	0.484	0.527
27. NEW: Project strategy	4.182	0.808	3.714	0.951	4.182	0.751	0.379	0.502
28. NEWEST: Project environment	4.121	0.820	4.429	0.787	4.091	0.944	0.655	0.645

In the table above, the p-value for each APSF for the ANOVA is larger than ($>$) 0.05. Some of the variables indicated that there was not homoscedasticity of variances and therefore a more robust test, the Welch test, was included. The Welch test's p-values for each of the APSFs are almost the same as that of the ANOVA's p-values and all the Welch p-values are also larger

than ($>$) 0.05. There is thus no statistically significant difference between the large, medium-sized and small organisations for all of the APSFs. This means that when comparing the averages of a specific APSF between the large, medium-sized and small organisations, it can be concluded that there is no statistically significant difference. All the respondents thus provided approximately the same average rating for a specific APSF.

Table 6.3: Sorted averages of the APSFs relevant to agile projects

Rank	Across all respondents	
	APSF	Mean
1	Management commitment	4.863
2	NEW: Communication	4.294
3	NEW: Resource management	4.294
4	Team capability	4.255
5	NEW: Collaborative teamwork	4.216
6	Client involvement	4.196
7	NEW: Monitoring and control	4.176
8	NEW: Testing	4.157
9	NEWEST: Project environment	4.157
10	Organisational environment	4.137
11	NEW: Role definition	4.118
12	NEW: Project strategy	4.118
13	Project definition process	4.098
14	NEW: Risk and issue management	4.039
15	Project management process	4.020
16	NEW: Change management and control	4.000
17	NEW: Reporting and tracking	4.000
18	Project schedule	3.922
19	Delivery strategy	3.902
20	NEW: Project selection and prioritisation	3.843
21	Project nature	3.843
22	Team environment	3.804
23	NEW: Project size	3.784
24	NEW: Technical designs, practices and tools	3.765
25	Agile software techniques	3.745
26	NEW: Client expectations	3.706
27	Project type	3.490
28	NEW: Politics and culture	3.431

The data in the table above represents the average of all fifty-one respondents and not the average of the separate means of large, medium-sized and small organisations. The averages in the tables are sorted in descending order. The top ten APSFs according to the fifty-one respondents across large, medium-sized and small organisation are presented in Table 6.3 above.

An interesting finding is that the newest identified APSF (see Table 6.1), *project environment* (identified in the analysed case studies), was in the top ten APSFs. Five of the top ten APSFs (*communication, resource management, collaborative teamwork, monitoring and control* and *testing*) were identified when the *agile project success criteria* were developed in Section 5.2.4, while four (management commitment, team capability, client involvement and organisational environment) were originally identified by Chow and Chao (2008:964). This confirms that six of the newly identified APSFs (see Table 6.1) are in the top ten APSFs rated as important to the success of agile projects by the fifty-one respondents.

While there were no statistically significant differences between the average ratings provided by the respondents in large, medium-sized and small organisations and although the lowest average rating is a 3, the APSFs with average ratings between 3 and 3.5 are given:

- 10: politics and culture;
- 23: project type.

A conclusion can be drawn that most respondents rated the APSFs, *project type*, and *politics and culture* less important than other APSFs relevant to the success of agile projects because across all respondents they had the lowest average rating.

A comparison of the top ten APSFs across all respondents given above with the top ten CPSFs in Table 5.1 of Kim *et al.* (2001:186) and the top ten CPSFs in Table 5.2 of Standish Group International (2001:4) is given in Table 6.4 below. The top ten APSFs across all respondents in Table 6.3 were prioritised, listed and numbered, and the CPSFs of Kim *et al.* and Standish Group International were grouped under the relevant APSF. Blank cells in the table indicate that a certain CPSF of Kim *et al.* (2001:186) or Standish Group International (2001:4) did not support a specific APSF.

Table 6.4: Comparison of the top ten APSFs with Kim et al. and Standish Group International

Rank	APSFs	Kim et al. (2001:186)	Standish Group International (2001:4)
1	Management commitment	4. (Lack of) Senior management support	1. Executive support
2	Communication		
3	Resource management		
4	Team capability	6. Project leader's (in)experience	3. Experienced project manager
5	Collaborative teamwork	3. (Lack of) Team-member commitment	
6	Client involvement	1. (Low) User participation in the project	2. User involvement
7	Monitoring and control	5. (Lack of) Project leader's project monitoring	
8	Testing		
9	Project environment		
10	Organisational environment	7. (Mis)Alignment of project and corporate goals 10. (No attempt to) Re-engineer business	

In Table 6.4, the four newly identified APSFs, *communication*, *resource management*, *testing*, and *project environment* of the top ten APSFs were not supported by Kim *et al.* (2001:186) or Standish Group International (2001:4). For clarity, the *team capability* APSF included project manager or project leader experience as a CPSF in Table 6.1, which is the reason that *project leader's (in)experience* (rated sixth by Kim *et al.*, 2001:186) and *experienced project manager* (rated third by Standish Group International, 2001:4) could be grouped under the *team capability* APSF in Table 6.4.

Seven of the top ten CPSFs of Kim *et al.* (2001:186) supported and were grouped under six of the top ten APSFs of question 2. Table 6.5 presents the grouping of the remaining three CPSFs of Kim *et al.* (2001:186) under the sorted averages of APSFs across all respondents (Table 6.3).

Table 6.5: Grouping of the remaining three CPSFs of Kim et al. under the prioritised APSFs

Rank	APSFs	Kim <i>et al.</i> (2001:186)
13	Project definition process	2. (Lack of) Clearly stated objectives
24	Technical designs, practices and tools	8. Use of (in)appropriate technology
18	Project schedule	9. (Lack of) A detailed project plan

In Table 6.5, *(lack of) clearly stated objectives* (rated second by Kim *et al.*, 2001:186) falls under the *project definition process* APSF, which was rated thirteenth across all respondents. *Use of (in)appropriate technology* (rated eighth by Kim *et al.*, 2001:186) falls under the newly identified APSF, *technical designs, practices and tools*, which was rated twenty-fourth across all respondents in Table 6.3. *(Lack of) a detailed project plan* (rated ninth by Kim *et al.*, 2001:186) falls under the *project schedule* APSF, which was rated eighteenth across all respondents in Table 6.3.

Only three of the ten CPSFs of Standish Group International (2001:4) supported and could be grouped under the top ten APSFs of question 2. Table 6.6 presents the grouping of the remaining seven CPSFs of Kim *et al.* (2001:186) under the sorted averages of APSFs across all participants (Table 6.3).

Table 6.6: Grouping of the remaining seven CPSFs of Standish Group International under the prioritised APSFs

Rank	APSFs	Standish Group International (2001:4)
13	Project definition process	4. Clear business objectives 5. Minimised scope
24	Technical designs, practices and tools	6. Standard software infrastructure
26	Client expectations	7. Firm basic requirements
15	Project management process	8. Formal methodology 9. Reliable estimates

In Table 6.6, *clear business objectives* and *minimised scope* (rated fourth and fifth, respectively, by Standish Group International, 2001:4) fall under the *project definition process* APSF, which was rated thirteenth across all respondents in Table 6.3. *Standard software infrastructure* (rated sixth by Standish Group International, 2001:4) falls under the *technical designs, practices and tools* APSF, which was rated twenty-fourth across all respondents in Table 6.3. *Firm basic requirements* (rated seventh by Standish Group International, 2001:4) falls under the *client expectations* APSF, which was rated twenty-sixth across all respondents in Table 6.3. *Formal methodology* and *reliable estimates* (rated eighth and ninth, respectively, by Standish Group International, 2001:4) both fall under the *project management process* APSF, which was rated fifteenth across all respondents in Table 6.3. Only nine CPSFs of the Standish Group International (2001:4) could be compared with the APSFs, as the tenth CPSF represented all *other* CPSFs.

Question 4: Extent to which the hybrid agile project management methodology (ver. 1) addresses the agile project success criteria (one-way ANOVA)

In question 4, the respondents from the large, medium-sized and small organisations indicated on a scale of 1 to 5 the extent to which they agreed that the hybrid APMM (ver. 1) addresses the twenty-eight APSFs in the *agile project success criteria*. The maximum score for full agreement was a 5, and the score for total disagreement was a 1. The output of the *one-way ANOVA* yielded the results given in Table 6.7 for the large, medium-sized and small organisations for each APSF.

Table 6.7: One-way ANOVA results for the extent to which the hybrid APMM (ver. 1) addresses the agile project success criteria

APSF	Large		Medium		Small		ANOVA	Welch
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	p-value	p-value
1. Management commitment	4.636	0.603	4.857	0.378	4.455	0.820	0.423	0.322
2. Organisational environment	4.091	0.723	4.000	0.577	4.000	0.632	0.904	0.898
3. Team environment	3.939	0.747	3.571	0.787	3.818	0.874	0.519	0.545
4. NEW: Project selection and prioritisation	3.909	1.071	3.429	0.976	3.545	0.934	0.397	0.405
5. Team capability	3.818	1.286	3.286	0.756	3.545	1.128	0.519	0.370
6. Client involvement	3.970	0.951	4.286	0.951	4.000	0.894	0.721	0.740
7. NEW: Role definition	4.273	0.876	4.143	0.900	4.364	0.809	0.870	0.875
8. NEW: Communication	4.424	0.663	4.143	0.690	4.273	0.905	0.596	0.603
9. NEW: Client expectations	3.970	0.883	4.000	1.000	4.182	0.603	0.772	0.684
10. NEW: Politics and culture	3.667	0.816	4.143	0.900	3.455	0.934	0.252	0.334
11. NEW: Collaborative teamwork	4.212	0.781	4.000	0.816	4.182	0.751	0.808	0.830
12. Project management process	4.242	0.830	4.143	0.900	4.273	0.467	0.939	0.941
13. Project definition process	4.212	0.820	4.143	0.378	4.455	0.688	0.598	0.483
14. NEW: Change management and control	4.303	0.810	4.143	0.900	4.455	0.688	0.717	0.721
15. NEW: Monitoring and control	4.394	0.704	4.143	0.900	4.545	0.688	0.525	0.615
16. NEW: Reporting and tracking	4.121	0.857	4.000	1.155	4.364	0.674	0.637	0.595
17. NEW: Risk and issue management	4.121	0.740	4.286	0.951	4.182	0.874	0.879	0.907
18. Agile software techniques	3.848	0.870	3.714	0.488	4.364	0.505	0.120	0.027
19. Delivery strategy	4.152	0.795	3.571	0.787	4.091	0.831	0.229	0.256
20. NEW: Technical designs, practices and tools	4.091	0.765	3.857	0.900	4.091	0.701	0.759	0.819
21. NEW: Testing	4.303	0.770	3.571	0.976	4.273	0.467	0.069	0.217
22. Project nature	4.273	0.719	3.857	1.069	4.091	0.701	0.401	0.548
23. Project type	3.879	0.781	3.714	0.951	4.091	0.539	0.571	0.515
24. Project schedule	4.000	0.791	3.714	0.488	4.000	0.775	0.652	0.458
25. NEW: Resource management	4.424	0.751	4.143	1.069	4.273	0.786	0.658	0.737
26. NEW: Project size	3.788	0.781	4.429	0.787	4.091	0.831	0.129	0.165
27. NEW: Project strategy	4.152	0.906	4.143	0.690	4.182	0.603	0.993	0.990
28. NEWEST: Project environment	4.061	0.864	4.286	0.951	4.455	0.688	0.386	0.342

In the table above, the p-value for each APSF for the *one-way ANOVA* test is larger than (>) 0.05. Some of the variables indicated that there was no homoscedasticity of variances and therefore a more robust test, the Welch test, was included. The Welch test's p-values for each of the APSFs are almost the same as that of the ANOVA's p-values and all the Welch p-values are also larger than (>) 0.05. There is thus no statistically significant difference between the large, medium-sized and small organisations for every APSF. This means when comparing the averages of a specific APSF for the large, medium-sized and small organisations, it can be concluded that they are not statistically significant different. All the respondents thus provided approximately the same average rating for a specific APSF regardless of the organisation's size.

Table 6.8: Sorted averages of APSFs addressed by the hybrid APMM (ver. 1)

Rank	Across all respondents	
	APSF	Mean
1	Management commitment	4.627
2	NEW: Monitoring and control	4.392
3	NEW: Communication	4.353
4	NEW: Resource management	4.353
5	NEW: Change management and control	4.314
6	NEW: Role definition	4.275
7	Project definition process	4.255
8	Project management process	4.235
9	NEW: Testing	4.196
10	NEW: Collaborative teamwork	4.176
11	Project nature	4.176
12	NEWEST: Project environment	4.176
13	NEW: Reporting and tracking	4.157
14	NEW: Risk and issue management	4.157
15	NEW: Project strategy	4.157
16	Organisational environment	4.059
17	Delivery strategy	4.059
18	NEW: Technical designs, practices and tools	4.059
19	Client involvement	4.020
20	NEW: Client expectations	4.020
21	Project schedule	3.961
22	Agile software techniques	3.941
23	NEW: Project size	3.941
24	Project type	3.902
25	Team environment	3.863
26	NEW: Project selection and prioritisation	3.765
27	Team capability	3.686
28	NEW: Politics and culture	3.686

The data in the table above represents the average of all fifty-one respondents and not the average of the means of the large, medium-sized and small organisations. The averages in the tables are sorted in descending order. The top ten APSFs that support the hybrid APMM (ver. 1) according to the fifty-one respondents across large, medium-sized and small organisation are presented in Table 6.8 above.

Six of the top ten APSFs that are addressed by the hybrid APMM (ver. 1) were also in the top ten APSFs relevant to agile projects. Seven of the top ten APSFs (*monitoring and control*, *communication*, *resource management*, *change management and control*, *role definition*, *testing* and *collaborative teamwork*) that are addressed by the hybrid APMM (ver. 1) were identified when the hybrid APMM (ver. 0) was developed (see Table 5.4), while the other three (*management commitment*, *project management process* and *project definition process*) were originally identified by Chow and Chao (2008:964). There is no APSF with an average ranking

of less than 3.5 when the averages of all respondents are considered (see Table 6.8). The conclusion can be drawn that the average respondent agreed that the hybrid APMM (ver. 1) addresses all of the APSFs because the averages of each APSF are larger than (>) 3.5 (midpoint between neutral and agree).

A concern however is that the *team capability* APSF is in the top ten for APSFs (across all respondents) that are relevant to agile projects (see Table 6.3), while it has the lowest average (together with *politics and culture*) across all participants for APSFs that are addressed by the hybrid APMM (ver. 1; see Table 6.8). This concern will be addressed in Section 6.3.2. Although *politics and culture* are the APSF with the lowest average rating for an APSF that is addressed by the hybrid APMM (ver. 1; see Table 6.8), this does not raise any concerns, as it is also the APSF with the lowest rating in question 2 (see Table 6.3).

Dependent t-test for the difference in means of Questions 2 and 4

The reason for conducting a paired t-test was to determine whether the APSFs that were rated by the respondents as relevant (and important) to agile projects in question 2 are addressed by the hybrid APMM (ver. 1) based on the respondents' opinions in question 4. Although there were no statistically significant differences between the average ratings provided by the respondents in large, medium-sized and small organisations in question 2 and 4, and although the average rating provided in question 2 and 4 across all participants was larger than 3.4, it was decided to investigate this difference at organisational level using a dependent t-test in small, medium-sized and large organisations. The reason for doing this was to be very thorough in the analysis in order to determine areas in which the hybrid APMM (ver. 1) could possibly be improved, by not only analysing the difference in average ratings for questions 2 and 4 across all participants, but also to analyse the difference in average ratings for questions 2 and 4 between small, medium-sized and large organisations (see large, medium and small group columns in Table 6.9). A dependent t-test was performed on the paired APSFs (see APSFs column in Table 6.9) to account for differences in confounding factors (for example, organisational policy, gender or age of respondent) within organisations. If a certain APSF was given a higher average rating in question 2 than it was given in question 4, it was raised as a concern to improve the hybrid APMM (ver. 1) in order to further address the identified APSFs – the dependent t-test determines the average difference between the two scores irrespective of the original scores.

Ellis and Steyn (2003:52–54) introduced effect size as a measure that “not only makes the difference independent of units and sample size, but relates it also with the spread of the data”.

The effect size will be calculated by $d = \frac{|\bar{x}_1 - \bar{x}_2|}{s}$ where s is the maximum of the two standard deviations from the samples. The effect size will be interpreted by the guidelines as provided by Cohen (1988:184–185):

- small effect: $d=0.2$;
- medium effect: $d=0.5$;
- large effect: $d=0.8$.

An APSF with $d \geq 0.8$ is considered to be practically significant, since it is the result of a difference having a large effect.

Table 6.9: Results of the paired t-test

APSFs	All groups			Large group			Medium group			Small group		
	Mean	Std. err. mean	Dif.	Eff. size	Mean	Std. err. mean	Dif.	Eff. size	Mean	Std. err. mean	Dif.	Eff. size
Q2_ Mngt Commitment	4.86	.056	0.235	0.37	4.85	.063	0.212	0.35	5.00	.000	0.143	0.38
Q4_ Mngt Commitment	4.63	.088			4.64	.105			4.86	.143		
Q2_ Org. Environment	4.14	.069	0.078	0.12	4.18	.092	0.091	0.13	4.00	.000	0.000	0.00
Q4_ Org. Environment	4.06	.095			4.09	.126			4.00	.218		
Q2_ Team Environment	3.80	.109	-0.059	-0.08	3.94	.130	0.000	0.00	3.29	.184	-0.286	-0.36
Q4_ Team Environment	3.86	.109			3.94	.130			3.57	.297		
Q2_ Proj. Selection & P...	3.84	.132	0.078	0.08	3.94	.162	0.030	0.03	3.86	.340	0.429	0.44
Q4_ Proj. Selection & P...	3.76	.144			3.91	.186			3.43	.369		
Q2_ Team Capability	4.25	.104	0.569	0.48	4.33	.120	0.515	0.40	4.00	.309	0.714	0.87
Q4_ Team Capability	3.69	.167			3.82	.224			3.29	.286		
Q2_ Client Involve...	4.20	.105	0.176	0.19	4.18	.141	0.212	0.22	4.14	.261	-0.143	-0.15
Q4_ Client Involve...	4.02	.130			3.97	.166			4.29	.360		
Q2_ Role Definition	4.12	.127	-0.157	-0.17	4.15	.152	-0.121	-0.14	4.14	.404	0.000	0.00
Q4_ Role Definition	4.27	.119			4.27	.152			4.14	.340		
Q2_ Communication	4.29	.126	-0.059	-0.07	4.39	.130	-0.030	-0.04	3.86	.508	-0.286	-0.21
Q4_ Communication	4.35	.100			4.42	.115			4.14	.261		
Q2_ Client Expect...	3.71	.135	-0.314	-0.33	3.61	.179	-0.364	-0.35	3.86	.261	-0.143	-0.14
Q4_ Client Expect...	4.02	.117			3.97	.154			4.00	.378		
Q2_ Politics and Culture	3.43	.123	-0.255	-0.29	3.48	.158	-0.182	-0.20	3.00	.309	-1.143	-1.27
Q4_ Politics and Culture	3.69	.120			3.67	.142			4.14	.340		
Q2_ Collaborative Team...	4.22	.106	0.039	0.05	4.27	.117	0.061	0.08	3.86	.459	-0.143	-0.12
Q4_ Collaborative Team...	4.18	.107			4.21	.136			4.00	.309		
Q2_ Pro. Mngt. Process	4.02	.099	-0.216	-0.28	4.06	.115	-0.182	-0.22	3.71	.286	-0.429	-0.48
Q4_ Pro. Mngt. Process	4.24	.107			4.24	.145			4.14	.340		
Q2_ Proj. Def. Process	4.10	.098	-0.157	-0.21	4.12	.121	-0.091	-0.11	3.71	.286	-0.429	-0.57
Q4_ Proj. Def. Process	4.25	.104			4.21	.143			4.14	.143		
Q2_ Change Mngt.&contr.	4.00	.115	-0.314	-0.38	3.97	.141	-0.333	-0.41	4.00	.309	-0.143	-0.16
Q4_ Change Mngt.&contr.	4.31	.110			4.30	.141			4.14	.340		

Table 6.9 continued

APSFs	All groups			Large group			Medium group			Small group		
	Mean	Std. err. mean	Dif.	Eff. size	Mean	Std. err. mean	Dif.	Eff. size	Mean	Std. err. mean	Dif.	Eff. size
Q2_ Monitor and Control	4.18	.124	-0.216	-0.24	4.15	.152	-0.242	-0.28	4.29	.286	0.143	0.16
Q4_ Monitor and Control	4.39	.101			4.39	.123			4.14	.340		
Q2_ Report and Track	4.00	.108	-0.157	-0.18	3.88	.136	-0.242	-0.28	4.29	.286	0.286	0.25
Q4_ Report and Track	4.16	.120			4.12	.149			4.00	.436		
Q2_ Risk & Issue Mngt.	4.04	.105	-0.118	-0.15	4.06	.137	-0.061	-0.08	4.00	.218	-0.286	-0.30
Q4_ Risk & Issue Mngt.	4.16	.110			4.12	.129			4.29	.360		
Q2_ Agile software Tech.	3.75	.118	-0.196	-0.23	3.82	.141	-0.030	-0.03	3.14	.404	-0.571	-0.53
Q4_ Agile software Tech.	3.94	.110			3.85	.152			3.71	.184		
Q2_ Delivery Strategy	3.90	.141	-0.157	-0.16	3.76	.169	-0.394	-0.41	3.71	.474	0.143	0.11
Q4_ Delivery Strategy	4.06	.113			4.15	.138			3.57	.297		
Q2_ Tech., practice & tool	3.76	.121	-0.294	-0.34	3.82	.147	-0.273	-0.32	3.43	.297	-0.429	-0.48
Q4_ Tech., practice & tool	4.06	.106			4.09	.133			3.86	.340		
Q2_ Testing	4.16	.117	-0.039	-0.05	4.15	.145	-0.152	-0.18	4.00	.309	0.429	0.44
Q4_ Testing	4.20	.109			4.30	.134			3.57	.369		
Q2_ Project Nature	3.84	.120	-0.333	-0.39	3.73	.152	-0.545	-0.62	3.86	.261	0.000	0.00
Q4_ Project Nature	4.18	.107			4.27	.125			3.86	.404		
Q2_ Project Type	3.49	.152	-0.412	-0.38	3.55	.180	-0.333	-0.32	3.43	.429	-0.286	-0.25
Q4_ Project Type	3.90	.106			3.88	.136			3.71	.360		
Q2_ Project Schedule	3.92	.125	-0.039	-0.04	3.97	.154	-0.030	-0.03	3.57	.297	-0.143	-0.18
Q4_ Project Schedule	3.96	.105			4.00	.138			3.71	.184		
Q2_ Resource Mngt.	4.29	.110	-0.059	-0.07	4.30	.141	-0.121	-0.15	4.14	.340	0.000	0.00
Q4_ Resource Mngt.	4.35	.111			4.42	.131			4.14	.404		
Q2_ Project Size	3.78	.106	-0.157	-0.19	3.76	.131	-0.030	-0.04	3.57	.297	-0.857	-1.09
Q4_ Project Size	3.94	.113			3.79	.136			4.43	.297		
Q2_ Project Strategy	4.12	.114	-0.039	-0.05	4.18	.141	0.030	0.03	3.71	.360	-0.429	-0.45
Q4_ Project Strategy	4.16	.113			4.15	.158			4.14	.261		
Q2_ Project Environment	4.16	.117	-0.020	-0.02	4.12	.143	0.061	0.07	4.43	.297	0.143	0.15
Q4_ Project Environment	4.18	.118			4.06	.150			4.29	.360		

Table 6.9 presents the APSFs that were paired together from questions 2 and 4. The APSF with at least a positive medium effect size ($d=0.5$) is highlighted in **red** in the table. This APSF will be addressed in Section 6.3.2 in seeking to improve the hybrid APMM (ver. 1) and is given below with the corresponding organisations (in brackets):

- team capability (medium and small).

This confirms the finding made earlier that *team capability* is in the top ten for APSFs (across all respondents) that are relevant to agile projects (see Table 6.3), while it has the lowest average in for APSFs (across all respondents) that are addressed by the hybrid APMM (ver. 1; see Table 6.8).

Where effect size (d) is negative (-) it means that the hybrid APMM (ver. 1) addresses a specific APSF to a greater extent than what was required by the respondents in questions 2. The APSFs with least a negative medium effect size ($d=0.5$) are highlighted in **green** in Table 6.9. The APSFs that are addressed to a greater extent by the hybrid APMM (ver. 1) than required according to the answers provided by respondents of large, medium-sized and small organisations are listed below with the corresponding organisations (in brackets) that support it:

- politics and culture (medium group);
- project definition process (medium group);
- agile software techniques (medium group, small group);
- project nature (large group);
- project type (small group); and
- project size (medium group).

From these findings, it is evident that there are far more APSFs that are adequately addressed by the hybrid APMM (ver. 1) than there are APSFs that are not adequately addressed. The answers to question 3 will be discussed in the section to follow.

Question 3: Other critical project success factors for agile projects

Most respondents did not respond to question 3 at all. Some of the suggested CPSFs provided by the respondents are not relevant to agile projects specifically. The CPSFs are more relevant to the strategic utilisation and implementation of the hybrid APMM (ver. 1) into an organisation. These factors were viewed as side factors and not as CPSFs. Some of the most important factors mentioned by the respondents across large, medium-sized and small organisations included the following:

- growth potential of the specific business environment – crucial to pre-empting within PMM;
- implement factors of differentiation – crucial to maintaining a competitive edge in the

competitive marketplace;

- clarification of the measures of success and KPIs, that is the implementation of new business strategies and the results measured accordingly; and
- employee training – to facilitate the change and the acceptance of an APMM.

Question 5: Suggested improvements for the hybrid APMM (ver. 1)

Most respondents also did not answer question 5, although some suggested some relevant improvements to the hybrid APMM (ver. 1), which should be considered:

- Adapt the methodology to place emphasis on realistic timelines, quality above quick delivery.
- Regularly conduct an environment scans to keep up to date regarding the latest technology in order to remain competitive, and to adapt regularly to ensure that quality and current products are delivered.
- Project vision should be re-visited after every phase and changes in processes, objectives and methodology could be required to accommodate the reaching of the final project vision (which should not be changed).

6.3.2 Discussion of statistical analysis results and suggested improvements to the hybrid APMM (ver. 1)

Project selection and prioritisation, team capability and politics and culture were the lowest ranked APSFs that have been addressed by the hybrid APMM (ver. 1; see Table 6.8).

Project selection and prioritisation was rated an average of 3.765 across all respondents (see Table 6.8), where the average respondent agreed that this APSF is addressed by the hybrid APMM (ver. 1). To improve the incorporation of this APSF by the hybrid APMM (ver. 1), it was moved to be the first key objective to be executed in the *pre-initiation phase*. This key objective will ensure that the organisation is executing the project with the highest priority after it has gone through the necessary selection and prioritisation procedures of the organisation.

Team capability, which arose as a concern based on the results of the *one-way ANOVA* and *paired t-test*, can only be addressed by appointing people with the required skills and expertise. This concern confirms the proposition made in theme 1 (*need the right people/resources on a team that is not short staffed with the right skills, allocated correctly and working together with the same vision – will result in a higher probability of project success*) in Section 6.2.3. In order to keep this simple, the concern was addressed in the same way as in Section 6.2.3, by emphasising the existing key objective: the *project manager(s) and team members appointed*

must have the relevant skills and experience to execute their responsibilities, in the pre-initiation phase of the hybrid APMM (ver. 2).

Although *politics and culture* is the APSF with the lowest average rating across all respondents that is addressed by the hybrid APMM (ver. 1; see Table 6.8), this does not raise major concerns, as the average respondent was not sure (average rating = 3.431) whether this APSF is relevant to agile projects. This finding confirms the proposition made in theme 3 (*human, behavioural, political and cultural factors are difficult to manage*, and *managing politics and culture*) and it was addressed in the same way as in Section 6.2.3, by treating people with respect (Holcombe & Holcombe, 2008:25). Furthermore, it is important that the project manager handle organisational politics instead of the team members becoming involved. If the major purpose and focus of everyone involved in a project is focused on delivering the project, politics can be minimised. It is also important for the project manager to manage meetings effectively by stipulating key outputs required beforehand to ensure that there are no fruitless discussions that could lead to negative political discussions.

In question 5, a respondent in one of the large organisations mentioned that he places “quality above quick delivery”. This statement was addressed by the hybrid APMM (ver. 1) in the *benefit realisation* step of the *iterative refactoring cycle* by ensuring that functionalities of quality and value are released during the *release new functionality* step. This will ensure that the project team not only delivers products quickly, but delivers products of quality as quickly as possible. If a deliverable of bad quality has been identified in the *improve and adapt* step in the *iterative refactoring cycle* it can be rectified or improved and tested again during the *test and evaluate* step as explained in Section 5.3.2.4.

One of the major observations made from the analysis of the answers to question 5 was that one respondent with sixteen years’ experience in project management suggested that a project and business environment scan be done regularly to ensure that the project is up to date on the latest technologies and trends. This will ensure that the organisation remains competitive and gain from using newer and better technologies than its competitors. The suggestion was addressed by the hybrid APMM (ver. 1) by adding environmental assessment as a key objective in the *adapt, direct, monitor and control* phase, and by adding it as an item to the agenda of the kick-off meeting. It was also added as an objective to be completed during *incremental development preparation* in order to ensure that the most current technology is utilised when developing deliverables.

A respondent in a small organisation emphasised in question 5 that the “project vision should be re-visited after every phase”. This was incorporated into the post-incremental meeting as an

agenda item of the hybrid APMM (ver. 1) in order to ensure that the team is always cognisant of the direction and vision of the project and organisation. It is important to ensure that the organisation's and project's visions are aligned with one another. If the vision is changed by an adaptive action the whole team and everyone directly and indirectly involved in the project must be notified immediately.

Other than the findings given above, there were no new findings identified from the answers to questions 3 and 5 that has not already been incorporated into the hybrid APMM (ver. 1). These improvements as explained above were incorporated in the hybrid APMM (ver. 1) to release the final hybrid APMM (ver. 2).

The changes and improvements made to the hybrid APMM (ver. 1) after the survey was completed will be explained in Section 7.3, where the final hybrid APMM (ver. 2) will be presented. All the solutions of the findings explained in this section were incorporated into the hybrid APMM (ver. 1) to create the hybrid APMM (ver. 2).

6.4 Summary

In this chapter, the evaluation of the hybrid APMM (ver. 0) developed in Chapter 5 was presented. The evaluation was based on the results gained from conducting case studies and a survey. Case studies were conducted at a large (case study A) and small (case study B) organisation, after which the case studies were analysed using content and cross-case analysis. The hybrid APMM (ver. 0) was improved by including the cross-case analysis findings, after which it was distributed with a questionnaire to different respondents in different organisations. The answers to the questionnaires were statistically analysed and the findings were incorporated into the hybrid APMM (ver. 1) to create the final hybrid APMM (ver. 2). The evaluation of the hybrid APMM (ver. 0 & 1) demonstrated that a combination of ASDMs and PMMs can be used to develop a hybrid APMM that has the ability to deliver IT projects successfully in a constantly changing business and project environment.

In Chapter 7, the findings of the study will be discussed in relation to the research objectives. The study's limitations and contributions will also be discussed. In addition, recommendations will be made for future work.

CHAPTER 7

FINDINGS AND FUTURE WORK

7.1 Introduction

This chapter will summarise the findings in relation to the six main research objectives of the empirical study: developing the *agile project success criteria*, the ASDM best practice framework, the PMM best practice framework and three versions of the proposed hybrid APMM (ver. 0, 1 and 2). This will be done in order to demonstrate that the research objectives, which was set in response to the research question in Chapter 1, has been achieved. Thereafter, the study will be concluded where the contributions and limitations will be detailed. The chapter will conclude with recommendations regarding future research to improve the hybrid APMM (ver. 2) further.

7.2 Research objectives and findings

The findings of the study will be reviewed in the sections that follow. The most pertinent findings of each research objective will be summarised. The discussion of the findings will give attention to the achievement of the research objectives of this study as stipulated in Chapter 1. Seven relatively well-established ASDMs and three PMMs were studied in Chapters 2 and 3 in order to gain a solid theoretical foundation of what could be used and combined in order to fulfil the research objectives of the study.

7.2.1 Develop the agile project success criteria

This objective was reported on in Chapter 5. After researching general CPSFs and general reasons that projects fail, they were summarised into CPSFs that represented the *critical project success criteria*. The reason for including general CPSFs, and not just factors for agile projects, were that in essence all projects have some agile elements because most of them are executed in a constantly changing environment. The summarised CPSFs of the *critical project success criteria* were grouped under the APSFs and categories of Chow and Chao (2008:964). The fifteen new APSFs identified were then combined with the twelve APSFs of Chow and Chao (2008:964) to create the *agile project success criteria*, which consisted of twenty-seven APSFs.

The propositions of theme 1 (general CPSFs), 2 (environmental characteristics) and 3 (people factors) identified through content and cross-case analysis was grouped and compared (in blue text) with the *agile project success criteria* (see Table 5.4) to identify any new APSFs in Table 6.1. Propositions of themes 2 and 3 were grouped under the APSFs of the *agile project success criteria* because they were just like theme 1, factors required for the successful

completion of a project. If a proposition could not be grouped under any existing APSF, it was identified as a new APSF which was presented in *green* text. The result of this comparison and grouping after the case studies were completed is presented in Table 7.1 (reproduced here from Table 6.1 for ease of reference). There was no other APSFs identified after the completion of the survey.

Table 7.1: Supporting CPSF grouping

Agile project success criteria	Supporting CPSFs
1. Management commitment	Executive/senior management ownership and support.
2. Organisational environment	- Adequate organisational change management. - Stakeholders, users and team members must understand the benefits of successful project delivery for the business. - Alignment of project and organisational objectives/vision. - Impact analysis of the project on the organisation. - Considering personal interests and ambition, but not placing these before the organisation's interests. - Understanding existing business processes. - Clearly defined and understood business drivers and objectives. - Attempt to re-engineer business.
3. Team environment	Effective, aligned and cross-functional team organisational structure.
4. NEW: Project selection and prioritisation	Projects must be selected based on an accurate and sound business plan.
5. Team capability	- Trained, experienced and skilled project managers and team members. - Good leadership. <i>Need a strong project manager/leadership (A2, A3, B1, B2, B3).</i> <i>Need the right people/resources on a team that is not short staffed with the right skills, allocated correctly and working together with the same vision – will result in a higher probability of project success (A1, A2, A3, B1, B2, B3).</i>
6. Client involvement	- User/client/stakeholder/sponsor involvement, participation, commitment, support, accountability and ownership. - Consultation with stakeholders/users/clients. - Considering stakeholders' employees' recommendations, ideas and solutions. <i>Keep stakeholders involved and provide regular feedback – this will ensure easier sign-offs (A1, A2, A3, B1, B2).</i>
7. NEW: Role definition	- Clearly defined roles, authority and responsibilities. - Understanding executive/senior management role in the project. <i>Stakeholder mapping (A1).</i>
8. NEW: Communication	- Constant and effective communication and feedback. - Willingness to share information. - Willingness to seek the advice of senior management. <i>Integrated communication (A1, A3, B1, B2).</i>
9. NEW: Client expectations	- Managing unrealistic client and business expectations. - Satisfied client, user, business and stakeholder expectations and requirements. - Clearly defined, gathered and controlled user and business requirements and expectations. <i>Stakeholders expect things to change immediately and they expect/force team members to deliver to unrealistic deadlines (A1, A2, A3, B1, B3).</i> <i>Users do not know what they want (A1).</i>
10. NEW: Politics and culture	- Managing politics and culture <i>Human, behavioural, political and cultural factors are difficult to manage (A1, A3, B1).</i>
11. NEW: Collaborative teamwork	- A project requires a collaborative effort. - Wellness and safety of team members are considered to be of high importance. - Motivated team members. - Quick and fair resolution of conflict. - Team-member commitment. - Self-controlled project team. <i>Know your stakeholders and team (A3).</i> <i>Project manager and team members must be on board to utilise and enforce a PMM (B1)</i> <i>Team members feel that the project manager does not always understand (B3)</i> <i>Performance management – people must be rewarded for doing a good job (A3, B1)</i> <i>People talk too much and do too little (B3).</i>
12. Project management process	- Adherence to high-quality standards and constraints for deliverables and the project. - Defined checkpoints, gate reviews and structure. - Correct application of project management theory. - Correct selection of the correct PMM. - Use and acceptance of the benefit of a defined, effective and standardised methodology. - Understanding the advantage of project management. - Processes and policies are defined with regard to project management best practices,

	approaches and concepts. - Knowing that project management does not happen at executive level. - Preventing scope creep - Reliable estimates. - Adequate cash flow management and fund provision. - Suitable and timely sign-offs. - Managing external project factors, such as vendor and regulatory issues. <i>Need adequate funding/budget (A1).</i> <i>Do benefit realisation regularly (A1, A2, A3, B1, B2, B3).</i> <i>Keep PMM/ASDM simple, minimise documentation (keep important documents) and focus on completing the work (A1, A2, A3, B1, B2, B3).</i> <i>Need a PMM/SDM that is flexible and agile – with the ability to adapt to change (A1, A2, A3, B1, B3).</i> <i>Need integrated project management and system development (A1, A3, B2).</i>
13. Project definition process	- Clearly defined, understood and controlled business case, benefits, objectives, deliverables and project definition. - Clear and unified vision/focus. - Defining and minimising a robust project scope and controlling scope changes. <i>Undefined/unclear scope (at beginning) – results in a great deal of rework (A1, A2).</i>
14. NEW: Change management and control	- Effective change identification, control, management, processes, and procedures. <i>Manage changing/unclear requirements – give stakeholders what they want, not what developers think they need (A2, B1, B2, B3).</i> <i>Inform people of changes and the project managers must ensure that the changes are enforced (A1).</i>
15. NEW: Monitoring and control	- Good quality control, management, processes, procedures, and acceptance criteria. - Understandable monitoring and controlling processes and procedures to track project progress. - Controlled project delivery. - Standardised reporting and monitoring procedures. - Resolving project problems identified as soon as possible. - Maintaining and updating the WBS. - Maintaining the project approval and sign-off processes. <i>Require proper monitoring and control procedure (A1, A2).</i>
16. NEW: Reporting and tracking	- Tracking and adhering to cost and budget estimates. - Standardised reporting and monitoring procedures. - Gathering lessons learnt as the project progresses. <i>Log and implement lessons learnt as the project progresses to prevent repeating mistakes (A1, A2, A3, B1, B2, B3).</i>
17. NEW: Risk and issue management	Regularly and promptly identifying, assessing, mitigating, resolving, reviewing, monitoring and controlling project risks and issues.
18. Agile software techniques	
19. Delivery strategy	- Use of prototyping. <i>Do maintenance after implementation (B1, B2, B3).</i> <i>Deliver quickly (A1).</i> <i>Formally close-out, sign-off and hand-over a project (A1, B2, B3).</i>
20. NEW: Technical designs, practices and tools	- Sufficient technical design, practices and utilisation of planning tools. - Performing a return on investment assessment. - No overreliance on technology. - Appropriate use and evaluation of, and experimentation with technology. - Standard software infrastructure.
21. NEW: Testing	High-quality testing.
22. Project nature	- Avoiding direction change in the middle of the project. - Considering the growth and evolution of a project. <i>Consider the nature of a project before applying a PMM/ASDM (A2, A3, B2, B3).</i>
23. Project type	
24. Project schedule	- Developing, tracking, adhering to, and correctly executing a detailed/robust project schedule and plan (verbal advice). - Acknowledgement that the project schedule and cost are inseparable. - Adequate budgets and realistic timelines.
25. NEW: Resource management (availability and allocation)	- Availability of resources and technical competencies (skills, hardware, software, sufficient funds, etc.). - Sufficient resources (money, time, expertise, etc.). - Proactive resource allocation. - Correct team and project manager selection and organisation. <i>Need the right people/resources on a team that is not short staffed with the right skills, allocated correctly and working together with the same vision – will result in a higher probability of project success (A1, A2, A3, B1, B2, B3).</i> <i>Ensure people are accountable for their tasks (A2).</i>
26. NEW: Project size	Sufficient project sizing.
27. NEW: Project strategy	- Appropriate project strategy. <i>Regular sign-offs are important, especially in a changing environment (A1, A2, A3, B2).</i>
28. NEWEST: Project environment	<i>Constantly fast-changing environment (dynamic) – change is the only constant (A1, A2, A3, B1, B2, B3).</i> <i>Project environment is challenging (A1).</i>

As explained in Chapter 6, from the twelve APSFs of Chow and Chao (2008:964), only six (delivery strategy, agile software-engineering techniques, team capability, project management

process, team environment, client involvement) were found by Chow and Chao (2008:964) to be relevant to agile projects, while there was no evidence for the other six APSFs as being prerequisites for the success of agile projects. After the CPSFs (identified from the completed case studies) were grouped under the twenty-seven APSFs in Table 7.1, it was found that they supported six of the twelve APSFs of Chow and Chao (2008:964). Four (delivery strategy, team capability, project management process, client involvement) of the six APSFs of Chow and Chao (2008:964) that are relevant to agile projects were supported by the CPSFs identified in the case studies.

After comparing the propositions of the case studies with the APSFs, two propositions from theme 2 (*constantly fast-changing environment (dynamic) – change is the only constant (A1, A2, A3, B1, B2, B3)* and *project environment is challenging (A1)*) in the case studies could not be grouped under the twenty-seven main APSFs. This resulted in a newly identified APSF called *project environment*, which was added to the *agile project success criteria*.

The updated *agile project success criteria* (Table 7.1) were utilised in questions 2 and 4 of the questionnaire in the survey to determine whether the twenty-eight APSFs are relevant to agile projects, and whether they are addressed by the hybrid APMM (ver. 1). The answers of question 2 of the questionnaire were statistically analysed through a one-way ANOVA. From the averages, it was found that the top ten APSFs according to fifty-one respondents across large, medium-sized and small organisations were the following:

- management commitment;
- communication;
- resource management;
- team capability;
- collaborative teamwork;
- client involvement;
- monitoring and control;
- testing;
- project environment; and
- organisational environment.

An interesting finding was that the newest identified APSF (see Table 7.1), *project environment* (identified in the analysed case studies), is in the top ten APSFs. Five of the top ten APSFs (*communication, resource management, collaborative teamwork, monitoring and control* and *testing*) were identified when the *agile project success criteria* were developed in Chapter 5, while four (management commitment, team capability, client involvement and organisational

environment) were originally identified by Chow and Chao (2008:964). This confirms that six of the newly identified APSFs are in the top ten APSFs rated as important to the success of agile projects by the fifty-one respondents.

It was found that most respondents rated the APSFs, *project type*, and *politics and culture* less important than other APSFs relevant to the success of agile projects because across all respondents they had the lowest average rating. There was no other APSFs identified after the completion of the survey.

A comparison of the top ten APSFs from the answers of question 2 (Section 6.3.1.1) with the top ten CPSFs of Kim *et al.* (2001:186; Table 5.1) and the top ten CPSFs of Standish Group International (2001:4; Table 5.2) found that seven of the top ten CPSFs of Kim *et al.* coincided with six of the top ten APSFs, while only three of the top ten CPSFs of Standish Group International coincided with the top ten APSFs.

The *agile project success criteria* is one of the first criteria developed that provides a comprehensive list of the APSFs that were found to be relevant to and important for the successful delivery of agile projects. This means that the ability to deliver an agile project will increase if these APSFs are addressed. PMMs and SDMs developed for agile projects in the future could utilise the *agile project success criteria* to ensure that their developed PMM or SDM has the ability to and higher probability of delivering projects successfully in a constantly changing project and business environment. Practitioners can utilise the *agile project success criteria* to increase the probability of delivering projects in a changing business and project environment or to determine which factors are not being addressed which might resolve the project related problems they might experience during project execution.

7.2.2 Develop the agile system development methodology best practice framework

This objective was also reported on in Chapter 5. The different ASDMs (DSDM, Scrum, XP, FDD, Crystal ASDMs, ASD, LD) studied in Chapter 2 were evaluated according to the *agile project success criteria* to develop the *ASDM best practice framework* (Table 5.5). The findings (strengths, weaknesses and gaps) on each APSF were used with the *PMM best practice framework* to develop the proposed hybrid APMM (ver. 0). The ASDM best practice framework is reproduced in Table 7.2 on the next page. The following evaluation responses were given:

- **(Y)**es: The ASDM/PMM addresses the APSF almost completely (81%-100%).
- **(P)**artially: The ASDM/PMM addresses the APSF partially (31%-80%).
- **(N)**o: The ASDM/PMM does not address the APSF (0%-30%).

Table 7.2: ASDM best practice framework

	Agile project success criteria	ASDMs							Findings
		DSDM	Scrum	XP	FDD	Crystal ASDMs	ASD	LD	
ORG.	1. Management commitment	P	P	P	P	P	P	P	Weakness. Senior management must treat the project as urgent and important; otherwise, there will be no commitment.
	2. Organisational environment	P	P	P	P	P	P	P	Weakness. The project must be regularly aligned with the organisation's focus, business processes and organisational change-management procedures.
	3. Team environment	P	P	P	P	P	P	P	Weakness. Team members must be correctly allocated to the project across different organisation divisions and the period of their allocation must be communicated to the rest of the organisation.
PEOPLE	4. NEW: Project selection and prioritisation	N	N	N	N	N	N	N	Weakness. Team members must be correctly allocated to the project across different organisation divisions and the period of their allocation must be communicated to the rest of the organisation.
	5. Team capability	N	N	N	N	N	N	N	Weakness. This can be addressed by employing people with the required skills and expertise.
	6. Client involvement	Y	Y	Y	Y	Y	Y	Y	Strength. Keeping the client involved is a fundamental value of all ASDMs.
	7. NEW: Role definition	P	Y	Y	Y	Y	P	P	Strength. Most ASDMs adequately define the role of every person involved.
	8. NEW: Communication	Y	Y	Y	Y	Y	Y	Y	Strength. Continuous and open communication is incorporated in all ASDMs.
	9. NEW: Client expectations	Y	Y	Y	Y	Y	Y	Y	Strength. The identification and satisfaction of client expectations are incorporated in all ASDMs.
	10. NEW: Politics and culture	N	N	P	N	N	N	N	Gap. XP addresses this with its respect value (Holcombe & Holcombe, 2008:25). Political and cultural issues can be managed by studying human behavioural sciences.
	11. NEW: Collaborative teamwork	Y	Y	Y	Y	Y	Y	Y	Strength. Collaborative teamwork is a fundamental requirement in all ASDMs.
PROCESS	12. Project management process	P	P	P	P	P	P	P	Weakness. This can be addressed through combining ASDMs with the best practices of project management.
	13. Project definition process	Y	Y	Y	Y	P	Y	P	Strength. Clearly defining what must be achieved by the project is incorporated in most ASDMs.
	14. NEW: Change management and control	P	Y	Y	P	P	Y	Y	Strength. Effective change management is a major focus of most ASDMs.
	15. NEW: Monitoring and control	Y	Y	Y	P	P	Y	Y	Strength. All ASDMs have either a monitoring or controlling function.
	16. NEW: Reporting and tracking	Y	Y	Y	P	P	Y	Y	Strength. Continual and rapid feedback/tracking is incorporated in most ASDMs.
	17. NEW: Risk and issue management	P	P	N	N	N	P	P	Weakness. Most ASDMs only incorporate risk management partially. This can be addressed through combining ASDMs with other methodologies that address risk and issue management.
TECHNICAL	18. Agile software techniques	Y	Y	Y	Y	Y	Y	Y	Strength. All of these ASDMs are agile software techniques in their own right.
	19. Delivery strategy	Y	Y	Y	Y	P	P	P	Strength. Continuous incremental integration is incorporated in all ASDMs.
	20. NEW: Technical designs, practices and tools	P	P	Y	P	P	P	Y	Strength. Part of most ASDMs, such as features, story cards, pair programming, delayed commitment, and JAD sessions.
	21. NEW: Testing	Y	Y	Y	Y	Y	Y	Y	Strength. Regular and integrated testing is incorporated in all ASDMs.
PROJECT	22. Project nature	P	P	P	P	Y	P	N	Strength. Crystal ASDMs show the potential to adapt to the nature of a project the best.
	23. Project type	P	P	P	P	Y	P	N	Strength. Crystal ASDMs show the potential to adapt to different project types the best.
	24. Project schedule	P	P	P	P	P	P	P	Weakness. This can be addressed through combining ASDMs with the best practices of project management.
	25. NEW: Resource management	P	P	N	N	N	P	N	Weakness. This can be addressed through combining ASDMs with the best practices of project management.
	26. NEW: Project size	P	P	P	P	Y	N	N	Strength. This can be addressed by Crystal ASDMs.
	27. NEW: Project strategy	P	P	P	P	P	P	P	Weakness. This can be addressed through combining ASDMs with the best practices of project management.
	28. NEWEST: Project environment	P	P	P	P	Y	P	P	Strength. This APSF was addressed using the framework of Crystal ASDMs and the agile project success criteria.

The findings presented in Table 7.2 demonstrated that the most weaknesses fell in the *organisational* and *project* categories. The analysis of these findings (Table 5.6) demonstrated that no single ASDM fully addresses the *agile project success criteria*. This emphasised the need for ASDMs to be combined with PMMs to create a hybrid APMM (ver. 0) that can better address the *agile project success criteria*. The identified weaknesses were adequately addressed by the hybrid APMM (ver. 0) and PMM best practice framework.

To be thorough the ASDMs in Table 7.2 and PMMs in Table 7.3 have been evaluated using the newest APSF: *project environment*. The findings of the evaluations were the same in both framework and they correspond with the solution incorporated into the hybrid APMM (ver. 2) to address the *project environment* APSF.

The framework presents the strengths and weaknesses of the seven ASDMs when applied to agile projects. This framework can thus be utilised by practitioners to select an ASDM for a project based on its strengths and weaknesses to ensure that the required APSFs are addressed by the ASDM selected. It can also be used to identify the key strengths and weaknesses before a certain ASDM can be combined with other methodologies to ensure that its strengths are utilised, its weaknesses are addressed and its gaps are bridged.

7.2.3 Develop the project management methodology best practice framework

This objective too was reported on in Chapter 5. The different PMMs (PMBOK, PRINCE2 and APM) studied in Chapter 3 were evaluated using the *agile project success criteria* to develop the *PMM best practice framework* (Table 5.7). The findings (strengths, weaknesses and gaps) for each APSF were used with the *ASDM best practice framework* to develop the proposed hybrid APMM (ver. 0). The PMM best practice framework is reproduced in Table 7.3 on the next page using the same evaluation responses as in Section 7.2.2.

Table 7.3: PMM best practice framework

	Agile project success criteria	PMMs			Findings
		PMBOK	PRINCE2	APM	
ORG.	1. Management commitment	P	P	P	<i>Weakness.</i> Senior management must treat the project as urgent and important; otherwise, there will be no commitment.
	2. Organisational environment	P	P	P	<i>Weakness.</i> The project must be regularly aligned with the organisation's focus, business processes and organisational change-management procedures.
	3. Team environment	P	P	P	<i>Weakness.</i> Team members must be correctly allocated across different organisation divisions and the period of their allocation must be communicated to the rest of the organisation.
PEOPLE	4. NEW: Project selection and prioritisation	N	N	N	<i>Weakness.</i> Team members must be correctly allocated to the project across different organisation divisions and the period of their allocation must be communicated to the rest of the organisation.
	5. Team capability	N	N	N	<i>Weakness.</i> This can only be addressed by employing people with the required skills and expertise.
	6. Client involvement	P	Y	Y	Strength. PRINCE2 and APM emphasise continual client involvement.
	7. NEW: Role definition	Y	Y	P	Strength. The roles of everyone involved are well defined in PMBOK and PRINCE2.
	8. NEW: Communication	Y	Y	Y	Strength. Continuous and open communication is incorporated in all PMMs.
	9. NEW: Client expectations	Y	Y	Y	Strength. The identification and satisfaction of client expectations are important in all PMMs.
	10. NEW: Politics and culture	P	P	P	Gap. This was addressed through combining with XP. Political and cultural issues can be managed by studying human behavioural sciences.
	11. NEW: Collaborative teamwork	Y	Y	Y	Strength. Collaborative teamwork is a fundamental requirement in all PMMs.
	12. Project management process	Y	Y	Y	Strength. Each PMM has a detailed project management process in place.
	13. Project definition process	Y	Y	Y	Strength. Clearly defining what must be achieved by the project is incorporated in all PMMs.
PROCESS	14. NEW: Change management and control	P	Y	Y	Strength. Effective change management is a major focus of PRINCE2 and APM.
	15. NEW: Monitoring and control	Y	Y	Y	Strength. All PMMs have a monitoring or controlling function.
	16. NEW: Reporting and tracking	Y	Y	Y	Strength. Continual and rapid feedback/tracking is incorporated in all PMMs.
	17. NEW: Risk and issue management	P	Y	P	Strength. Risk and issue management is fully incorporated by PRINCE2.
	18. Agile software techniques	N	P	Y	Strength. APM is the only PMM that fully incorporates this factor.
TECHNICAL	19. Delivery strategy	P	Y	Y	Strength. PRINCE2 and APM support continual incremental integration.
	20. NEW: Technical designs, practices and tools	Y	Y	Y	Strength. All PMM has well-defined practices executable in practice.
	21. NEW: Testing	P	Y	Y	Strength. Testing is integrated across all areas of PRINCE2 and APM.
PROJECT	22. Project nature	P	Y	Y	Strength. PRINCE2 and APM show the potential to adapt to the nature of a project the best.
	23. Project type	P	Y	Y	Strength. PRINCE2 and APM show the potential to adapt to different project types the best.
	24. Project schedule	Y	Y	P	Strength. PMBOK and PRINCE2 clearly define this APSF.
	25. NEW: Resource management	Y	Y	P	Strength. PMBOK and PRINCE2 clearly define this APSF.
	26. NEW: Project size	N	Y	Y	Strength. PRINCE2 and APM show the potential to adapt to different project sizes the best.
	27. NEW: Project strategy	Y	Y	Y	Strength. All PMMs define a strategy to be followed to manage a project.
	28. NEWEST: Project environment	P	P	P	Strength. This APSF was addressed using the framework of Crystal ASDMs and the <i>agile project success criteria</i> .

The findings presented in Table 7.3 demonstrated that the most weaknesses fell in the *organisational* categories. The analysis of these findings (Table 5.6) demonstrated that no single PMM fully addresses the *agile project success criteria*. This emphasised the need for PMMs to be combined with ASDMs to create a hybrid APMM (ver. 0) that can better address the *agile project success criteria*. The identified weaknesses were adequately addressed by the hybrid APMM (ver. 0).

In both frameworks, *politics and culture* was identified as a gap. The suggested solution for addressing this APSF is to treat people with respect and to honour each other's viewpoints, culture, religion and political understanding.

The framework can be used for the same functions as the ASDM best practice framework. The PMM best practice framework presents the strengths and weaknesses of three PMMs when utilised for agile projects. This framework can thus be utilised by practitioners to select a PMM for a project based on its strengths and weaknesses to ensure that the required APSFs are addressed by the PMM selected. It can also be used to identify the key strengths and weaknesses before a certain PMM can be combined with other methodologies to ensure that its strengths are utilised, its weaknesses are addressed and its gaps are bridged.

7.2.4 Develop a proposed hybrid APMM (ver. 0) that would address the agile project success criteria

This objective was reported on in Chapter 5. In order to address the need for an APMM, the hybrid APMM (ver. 0) was developed using design science (research approach) and constructivism. This was done by merging different components of ASDMs and PMMs and by combining the strengths, addressing the weaknesses and bridging the gaps of the ASDMs and PMMs (see Tables 7.2 and 7.3) to ensure that the *agile project success criteria* were addressed. In order to show where the *agile project success criteria* were addressed by the hybrid APMM (ver. 0), the number assigned to each APSF (Tables 7.2 and 7.3) was placed behind the different sentences that addressed a specific APSF. By addressing the *agile project success criteria*, the hybrid APMM increased its ability to delivering IT projects successfully in a constantly changing environment.

7.2.5 Develop an improved hybrid APMM (ver. 1) by conducting two case studies

This objective was reported on in Chapter 6. Case studies were conducted at a large (case study A) and small (case study B) organisation. After sending the hybrid APMM (ver. 0) to the three participants at each organisation, interviews were conducted and transcribed. The content of the interviews was analysed and the propositions of each case study were summarised under six themes. The results of the two case studies were then compared using cross-case analysis (Section 6.2.2) and the propositions were grouped under the same six themes. The results of the case studies were used to effect changes and improvements to the hybrid APMM (ver. 0), resulting in the hybrid APMM (ver. 1). The changes and improvements made to the hybrid APMM (ver. 0) after the case studies had been completed will be explained in Section 7.3.

7.2.6 Develop an improved hybrid APMM (ver. 2) by conducting a survey

This objective was reported on in Chapter 6. After developing the hybrid APMM (ver. 1), it was distributed together with a questionnaire of five questions to sixty-five selected respondents of large (>100 employees), medium (50–100 employees), and small (<50 employees) organisations in order to gather quantitative and qualitative data. As explained in Section 7.2.1, the updated *agile project success criteria* (Table 7.1) were used in questions 2 and 4 of the questionnaire to determine whether the twenty-eight APSFs are relevant to agile projects, and whether they are addressed by the hybrid APMM (ver. 1). After collecting fifty-one (78.4%) questionnaires, they were statistically analysed using *one-way ANOVA* of the answers to questions 2 and 4 and a *paired t-test* of the answers to questions 2 and 4. The results of the statistical analyses (including the findings of questions 3 and 5) were used to effect improvements to the hybrid APMM (ver. 1), resulting in the hybrid APMM (ver. 2). The changes and improvements made to the hybrid APMM (ver. 1) based on the results of the survey will be explained in Section 7.3.

The data received from the respondents' questionnaires was statistically analysed to determine to which extent the hybrid APMM (ver. 1) addressed the *agile project success criteria*. The conclusion was drawn that the hybrid APMM (ver. 1) addressed all of the APSFs because the average ranking provided by the respondents for each APSF in the *agile project success criteria* was larger than (>) 3.5 (midpoint between neutral and agree). It was furthermore determined from the dependent t-test (Section 6.3.1.1) that the hybrid APMM (ver. 1) addressed certain APSFs in the *agile project success criteria* far more than required by the average participant.

7.3 Contribution and conclusion of the study: Hybrid Agile Project

Management Methodology (ver. 2)

In this section the effectiveness and development of the final hybrid APMM (ver. 2) from its first version is explained, which is the main contribution of this study. This hybrid APMM (ver. 2) has been the first methodology developed using the strengths, addressing the weaknesses and bridging the gaps of three PMMs and seven ASDMs. Utilising the hybrid APMM (ver. 2) could increase the probability of delivering IT projects in a constantly changing environment because it was found to be able to do so.

Any project manager who would like to execute and deliver an IT project in a constantly changing business or project environment could utilise the hybrid APMM (ver. 2). This hybrid APMM (ver. 2) will support practitioners by providing them with a simple and effective way to manage and deliver IT projects in which change is inevitable. The hybrid APMM (ver. 2) has the same advantages as the components of the different PMMs and ASDMs from which it was developed and it addresses all the APSFs of the *agile project success criteria*.

Since the case studies and survey revealed the effectiveness of the hybrid APMM (ver. 0 and 1), it can be concluded that a combination of ASDMs and PMMs can be used to develop a hybrid APMM that has the ability to deliver IT projects successfully in a constantly changing business and project environment. After evaluating the hybrid APMM (ver. 0 and 1) it was found to be acceptable, implementable, flexible, simple, fresh and new. It has the ability to adapt to constant change and it creates a culture in which people can embrace change instead of resisting it. It minimises documentation, keeps stakeholders involved, and focuses on completing the work. It furthermore ensures a quick response and resolution of identified problems, issues and risks. It also allows teams to achieve objectives more quickly and it will be able to deliver IT projects within time, scope, quality and cost constraints.

The original hybrid APMM (ver. 0) is discussed in black text. The changes and improvements made to the hybrid APMM (ver. 0) after the completion of case studies A and B to form the hybrid APMM (ver. 1) is presented in blue text. The changes and improvements made to the hybrid APMM (ver. 1) after the completion of the survey to form the hybrid APMM (ver. 2) will be given in maroon text.

Summary of hybrid APMM (ver. 2) reflecting changes and improvements

The hybrid APMM (ver. 2) life cycle, which consists of seven phases, is presented in Figure 7.1. The new hybrid APMM (ver. 2) is simplified by only providing phases and explains the integration amongst the phases, instead of using endless process flows and activity lists.

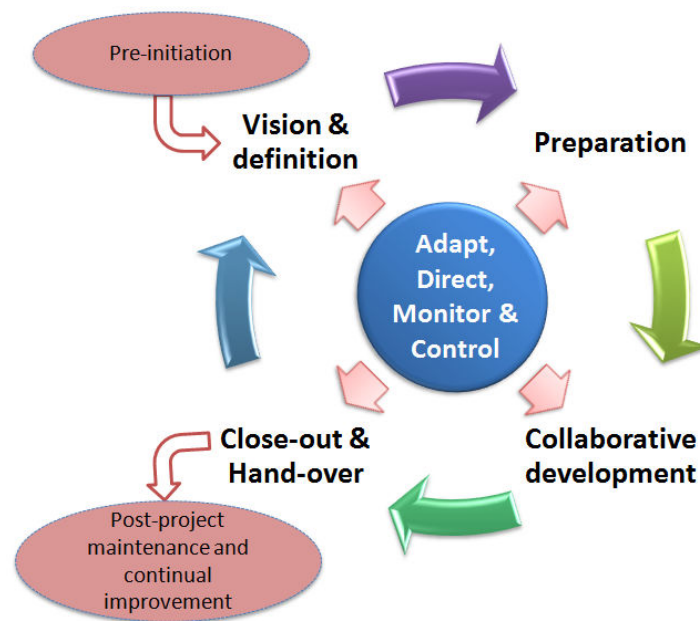


Figure 7.1: The proposed hybrid APMM (ver. 2) life-cycle phases

Each phase will be explained in detail below with regard to the functionalities and responsibilities of the different phases.

Pre-initiation phase

The *pre-initiation* phase ensures that everything is in place for the project to be initiated successfully. The key objectives of this phase include the following:

- The project with the highest priority and importance within an organisation should be the project that must be executed first. This selection and prioritisation of projects is normally completed by the portfolio management function of an organisation through which management ensures that the organisation's strategic objectives are achieved by the prioritised projects that must be executed. This ensures that the projects in execution are aligned with the strategic direction of the organisation.
- The initiation of the project is justified in a business case that defines a preliminary project scope, or a preliminary feasibility study is completed to check whether it would be feasible to commence the project.
- The necessary senior management and stakeholders are assigned to provide approval on collaborative increments completed.
- The different ways in which the project can be developed are evaluated and the approach to developing the project collaboratively is identified.
- People are appointed as part of the project team or as project manager(s) who will be responsible for executing the work required as stipulated in the project's *vision and definition*. When individuals are appointed from different divisions within an organisation, it

is important to ensure that team members are correctly allocated and that the time and duration of their allocation to a project are communicated to rest of the organisation. This will ensure that resources are not over-utilised within an organisation. The project manager(s) and team members appointed must have the relevant skills and experience to execute their responsibilities and to apply the assigned PMM correctly. A strong project manager(s) increases the probability of delivering a project successfully.

- A decision is made regarding whether the project must commence. Time is not spent on initiating a project that is based on unreliable assumptions and expectations regarding scope, time, cost constraints and acceptance criteria. The project must only commence if senior management view the project as important and they must give the assurance of and commitment to providing full support and guidance for the duration of the project.
- The tasks to be completed during the *vision and definition* phase are *speculated on*.
- The nature of the project must be defined by identifying unique characteristics of a project in order to ensure that the hybrid APMM (ver. 2) is the correct methodology to be used for the completion of the project. The unique characteristics will be identified by estimating how well the twenty eight APSFs of the *agile project success criteria* are addressed by the project. Questions related to each APSF will be asked to determine the characteristics, complexity and uniqueness of the project. The project/system critically will also be defined using the framework of Crystal ASDMs (see Figure 2.11), which would contribute to the uniqueness of the project. These unique characteristics will be used to determine whether the hybrid APMM (ver. 2) is the correct approach to commence the project. If the nature of a project presents problems for the utilisation of the hybrid APMM (ver. 2), it would be wise to assess whether the hybrid APMM (ver. 2) should be combined with other methodologies, or to use another methodology altogether.
- Assurance from the stakeholders must be received, preferably as a signed document stating that the organisation has the necessary funding to commence and complete the project after taking contingencies, possible project changes, and project risks into consideration.

The business case or feasibility study will then be used as a base-line to develop the project charter in the *vision and definition* phase.

Vision and definition phase

The project must have a vision that defines what has to be achieved by when, in order to determine during the *close-out and hand-over* phase whether the project or programme has been delivered successfully. The vision ensures that everyone associated with the project understands the objectives and his/her responsibilities in order for everyone to work towards the

same objective as a whole. During this phase, the project vision, preliminary scope, community and *collaborative development* approach (tailored as necessary) are determined. The key objectives of this phase include the following:

- stipulating the project's vision – what has to be achieved;
- defining the required key capabilities and objectives;
- in order to do benefit realisation in the *collaborative development* phase, the benefits expected, related risks, issues and the reason for executing the deliverables must be defined;
- speculating on the time-frame and cost of developing the objectives of the project collaboratively;
- identifying who makes the necessary decisions and provides approval;
- defining the manner in which quality, risks and progress will be measured, controlled and tracked.
- defining an effective *integrated* communication structure and levels of authority;
- tailoring the hybrid APMM (ver. 2) further to suit the project environment;
- speculating on the activities to be completed during the *preparation* phase;
- empowering the team to ensure that the objectives and capabilities required are delivered successfully;
- ensuring that the vision of the project is aligned with the vision of the organisation. (If the project's vision is not aligned with the organisation's vision, it must be re-evaluated and adjusted to ensure that it addresses the organisation's expectations as well.); and
- *setting up a stakeholder map to ensure that the stakeholders are identified and kept updated regarding the progress of the project.*

In order to adhere to the principle of keeping documentation to a minimum and rather focusing on completing the work, it is necessary to constantly determine the bare essentials of the process and documentation needed. There must be a focus on keeping it as simple as possible and not spending time on developing lengthy preliminary project plans and schedules, which are likely to change anyway.

The document that will be completed in this phase is the *project charter*, which will use the selected solution in the *feasibility study*, or the *business case*, which was completed in the *pre-initiation* phase, together with the key objectives mentioned above, as the base-line. In most cases, the same document used for the *business case* can just be updated after the vision and project's definition have been clarified. The document will justify the project, put major processes of understanding in place, and identify the main stakeholders for the project. Furthermore, it will define the project's focus, major objectives and deliverables and state by

whom they must be delivered within the associated deliverable cost, time and quality constraints.

Preparation phase

The team must proactively speculate on or anticipate the unknown, while they plan for what is definite. They can thus proactively prepare for what is known and what is unknown in the *preparation* phase.

Although uncertainty cannot be eliminated through planning – a person can prepare to adapt by anticipating what could happen should the uncertainty occur, or even before it occurs. For example, when a large project team has to move offices, it can be assumed that it will take the team time to settle into its new environment and that deliverables will be completed more slowly than usual. The same is true for projects; based on experience, a project manager or team member can predict or anticipate what may happen next based on the current circumstances and project environment. He/she can then proactively prepare beforehand regarding the manner in which to react when it something does or does not happen.

The key objectives of this phase include:

- gathering as many requirements as possible beforehand;
- defining the workload by grouping and prioritising deliverables, deliverables sets and incremental release plans;
- developing an incremental release plan and project schedule guideline;
- incorporating risk mitigation and issue resolution strategies into the plan; and
- speculating on and estimating project costs, timelines, roles and execution responsibilities.

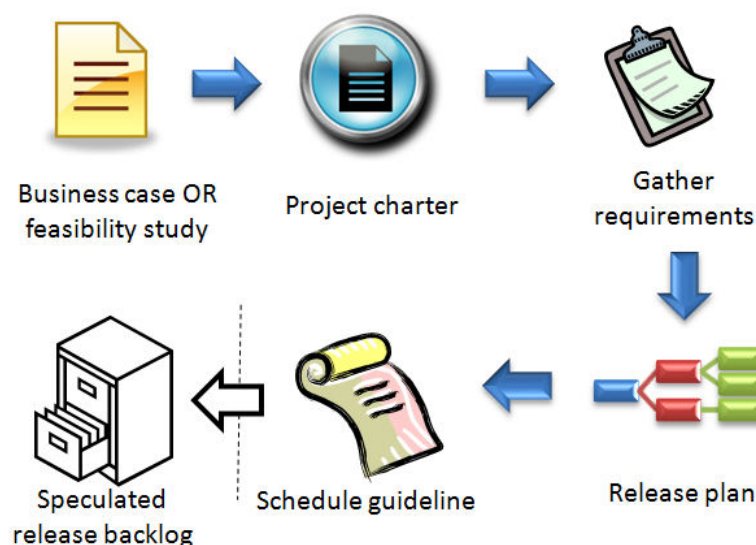


Figure 7.2: The preparation phase

In Figure 7.2 above, the *project charter* that contains the vision and definition of the project is used to gather as many user and business requirements as possible beforehand. The reason that as many requirements as possible must be gathered before collaborative development takes place is to decrease the possibility of new and changing requirements. Gathering as many requirements as possible beforehand will also ensure that planning and speculation are done based on more known requirements and facts than unknown requirements, which will result in a more accurate and *relevant project schedule guideline*. The project schedule guideline consists of assumptions and speculations that will result in the prioritised *speculated release backlog*, which will be the starting point of the *collaborative development* phase.

Requirements are gathered through workshops, JAD sessions, questionnaires or story cards. The requirements are then analysed and explained as *deliverables* and *activities* to be completed in order to satisfy the required client expectations. In Figure 7.3, the defined *deliverables* and *activities* are grouped into logical *deliverables sets* after the *deliverables* have been prioritised based on their level of importance. The *deliverables sets* are then grouped into *incremental release plans* after they have been prioritised based on their level of importance. The *incremental release plans* are then also prioritised after which the one with the highest priority is selected to be executed as the next *collaborative increment* in the *collaborative development* phase. Each *collaborative increment* is time-boxed.

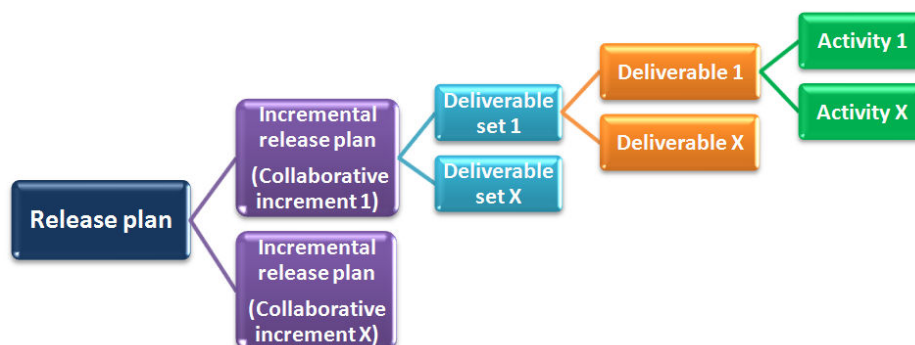


Figure 7.3: The release plan

In order to adapt to the different levels of complexity in projects, one *deliverable* can be completed as part of a *deliverables set*, and one *deliverables set* can even be completed as an *incremental release plan*. This means that one *deliverable* can be viewed as an *incremental release plan* in small projects.

After grouping and prioritising activities, deliverables, deliverables sets and incremental release plans, the release plan is used to create the project schedule guideline. The project schedule guideline will contain the typical WBS of who does what, how much each deliverable will cost and by when a certain deliverable must be completed. The *scheduled guideline* will be used to

guide the project as a whole by keeping the *vision* of the project in mind. Some deliverable dates, deliverables as a whole, and persons responsible for completing deliverables will change as the project progresses. For this reason, it is not called a scheduled plan but a *scheduled guideline*. The *scheduled guideline* will then be used as the speculated release backlog in the *collaborative development* phase.

Customers are involved during every step of the preparation phase to ensure that the team understand the requirements and expectations, and to ensure the client understand the threats and constraints the project team might be facing. The collaborative phase is directly dependant on the preparation phase.

Collaborative development phase

The reason that this phase is called *collaborative development* is that the word “collaborative” implies that teamwork, communication and constant stakeholder involvement are very important to ensure the successful development and delivery of each *collaborative increment* and the project as a whole. Clients are involved and consulted at every step, and preparation is integrated throughout the collaborative development phase. The key objectives to be completed during this phase are explained in the sections that follow.

Iterative refactoring cycle

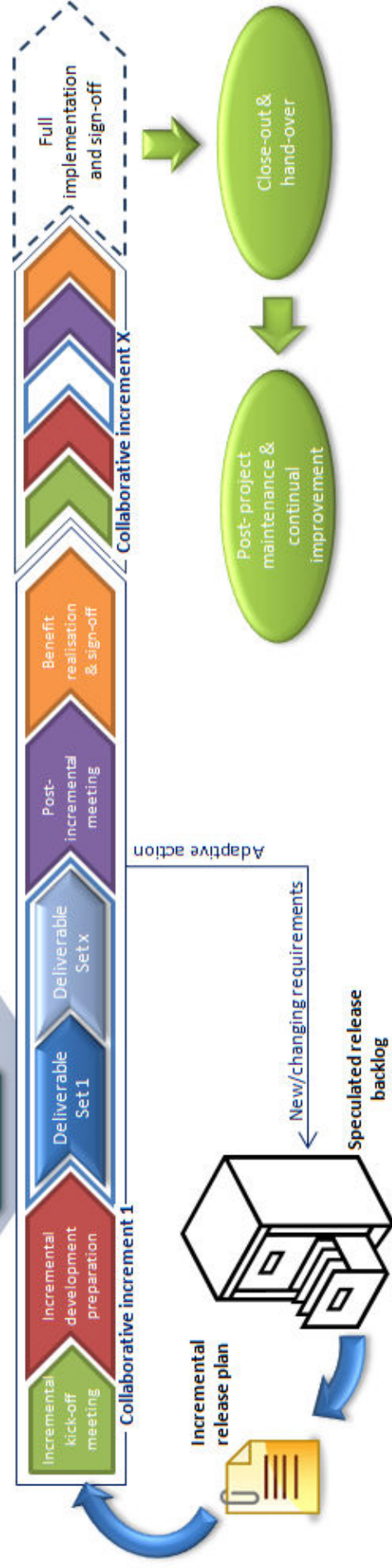


Figure 7.4: The collaborative development phase

The collaborative development phase is presented in detail in Figure 7.4. The phase starts with the *speculated release backlog*, which contains prioritised incremental release plans. When an *incremental release plan* with the highest priority is selected, it becomes a *collaborative increment*, which will be executed and signed off to release a new functionality of value that satisfies client objectives and expectations.

Every *collaborative increment* is commenced by a kick-off meeting. Everyone involved in executing and approving the *deliverables set(s)* and *collaborative increment* are invited to this meeting in order to ensure that effective *incremental development preparation* is completed by defining the manner in which the deliverables will be executed by whom and by when. An *environmental assessment* is also completed to ensure that the project is up to date on the latest technologies and trends. This will ensure that the organisation remains competitive and gains from using newer and better technologies than its competitors. The reason for doing preparation again is that some changes (resources, deliverables, timelines, *technologies*, *tools*, etc.) might have occurred since the *project schedule guideline* was created in the *preparation* phase, and it is also done to make sure that the project's vision is still aligned with the organisation's vision and objectives. It is important that preparation and speculation happen at a project level during the *preparation* phase and at a *collaborative incremental* level in order to ensure that all new and changed constraints and expectations are taken into consideration before the *deliverables sets* are executed.

After completing *incremental development preparation*, every *deliverables set* is executed using the *iterative refactoring cycle*, in which the design is improved until it is of *good quality and* value. The *iterative refactoring cycle* can be run through more than once for a specific *deliverables set*, or for every *deliverable* in the *deliverables set*. More than one deliverable can thus run simultaneously through the *cycle*. The *cycle* commences with a *speculate and analyse* step to ensure that if any changes occurred they are taken into consideration while the project has progressed. The necessary analyses are also completed before the deliverable(s) is physically *created* (or designed).

The deliverable(s) is *evaluated and tested* after the *create* step has been completed. Here the project manager and project team members determine whether the promises originally made have been fulfilled and thus that the expectations of the users have been met. If the deliverable(s) does not meet the acceptance and quality criteria, the deliverable(s) is improved during the *improve and adapt* step and tested again during the *test and evaluate* step. Even if the deliverable passed the test and evaluate phase the first time, it is important to check again whether the deliverable cannot be improved or adapted even more within the allowed time and cost constraints to deliver an even better deliverable(s) that would ensure absolute client satisfaction. During these steps, the unique

characteristics (defined in the *pre-initiation* phase) of the project must also be reviewed to ensure that the nature of the project has not changed. In a situation where a drastic change has to be made to the deliverable(s) because of the project's nature that changed, or because of *new or changing requirements*, an *adaptive action* must be executed to rectify the problem by including the new or changed requirement(s) in the *create* step, which must be executed again together with the remainder of the step in the *iterative refactoring cycle*. The *improve and adapt* phase also ensures that functionalities are included by the project team in a deliverable(s) that clients have not identified and that may result in a failed deliverable(s).

After the completion of the *improve and adapt* step, the deliverable(s) are reviewed in the *benefit realisation* step to determine whether the value expected in accordance with the *project schedule guideline* and the client expectations has been realised. In many projects, deliverables are completed successfully but they do not bring value to the organisation or to the project as a whole. The *benefit realisation* step in the *iterative refactoring cycle* ensures that functionalities of good quality and value are released during the *release new functionality* step. This will ensure that the project team not only delivers products quickly, but also delivers products of quality as quickly as possible. The functionalities can be released in the form of prototypes.

Learning is amplified in the *iterative refactoring cycle* because as deliverables progress through the different cyclical steps many lessons are learnt that must be logged on a regular basis. In the *learn* step, the project team can depend on previous experiences and solutions to problems to ensure that the same mistakes are not repeated during the execution of the next *deliverables set*. The lessons learnt can also be used during *incremental development preparation* to ensure that future *collaborative increments* do not make the same mistakes as in the past. The lessons learnt can further be utilised at a project level when the *preparation* phase is commenced for a new project in order to ensure that the mistakes made in previous projects are prevented in future projects.

During the *iterative refactoring cycle*, short *daily stand-up meetings* are conducted. The reason for having stand-up meetings is to ensure that people do not talk and waste time by discussing matters that are not urgent or relevant to the project when they sit down. During this meeting, problems and project changes are discussed, quick solutions are found, and assistance is provided by the project team if necessary. Furthermore, it ensures that the team members stay focused on the vision and the task at hand. Issues and risks are quickly resolved to ensure that any negative impact on the project is minimised.

The *direct, monitor and control* process of the *iterative refactoring cycle* is integrated with the *adapt, direct, change and control* phase. The only difference is that this process occurs at a deeper

iterative level of the project and not at an overall project level. This process directs, monitors and controls the *iterative refactoring cycle* as deliverables progress through the steps of the cycle. Progress, issues, risks, problems and new functionalities are tracked and reported upon to all stakeholders. Delayed commitment is applied within the *iterative refactoring cycle* to postpone irreversible decisions for as long as possible without affecting the cost, time and quality constraints until informed decisions can be made based on definite facts instead of predictions.

After the completion of the *deliverables sets*, a *post-incremental meeting* is conducted, in which the project team, clients and other stakeholders go through a final *benefit realisation* exercise after all the deliverables sets have been completed within a specific *collaborative increment*. During the *post-incremental meeting*, all the changes that occurred in the increment must be discussed and the project manager must ensure that the changes have been implemented or enforced. In the *post-incremental meeting*, the project vision should be re-visited to ensure that the team is always cognisant of the direction and vision of the project and organisation. It is important to ensure that the organisation's and project's visions are aligned with each other. If the vision was changed by an adaptive action, the whole team and everyone directly and indirectly involved in the project must be notified immediately (via an organisation communiqué).

If the *deliverables sets* consist of only one deliverable each, it will not be necessary to go through another *benefit realisation* exercise, as this would have been completed during the *iterative refactoring cycle*. The *collaborative increment* is approved and signed-off when all the client expectations have been satisfied, benefits have been proven, and value has been realised, as promised in the *project schedule guideline* and the *incremental development preparation* step. Collaborative incremental delivery is done by having regular sign-offs. During the *benefit realisation and sign-off*, it is important to check that what has been delivered is still aligned with the organisation's vision.

An important note of caution is that regular sign-offs can be counterproductive if not managed correctly. The suggestion is to have incremental sign-offs when a collaborative increment has been completed for large projects. If a small to medium-sized project's collaborative increment has been completed, a sign-off is not required to continue with the following collaborative increments. The advice however will still be that regular sign-offs must be done where a set of collaborative increments can be combined (dependable on the size of the project) into one sign-off to ensure that the client does not later claim that a set of deliverables completed was not what they required. If the project is very small, only one sign-off is required when the project is completed. It must also be clearly stipulated and understood by team members that they may not wait for sign-offs to continue with their work, unless a direct order to stop work has been given by the stakeholders or

project manager. Whether it is a large or small project, the next collaborative increment can be commenced without a sign-off of previous dependable increments. If there are changes, they can always be addressed with the change-control procedures and adaptive actions embedded throughout the hybrid APMM (ver. 2). After the post-incremental meeting has been concluded and the increments have been signed-off in large projects, the progress made should be calibrated by having a social/celebration/team-building session, where everyone involved can connect at a social level. For small projects, a social event should only be organised to celebrate the successful completion of the project, while regular team-building activities (if the funds are available) should be organised to allow team members to connect at a social level.

During *benefit realisation and sign-off* at a *collaborative increment* level, *new or changing requirements* and expectations can arise that must be added to the *speculated release backlog*. If a significant or large change occurred during the *approve and adapt step* that could not be solved during the *iterative refactoring cycle*, this may result in a *new or changing requirement*, which must be added by an *adaptive action* to the *speculated release backlog*. After any new or changing requirements have been added to the *speculated release backlog*, these must quickly be regrouped and prioritised before the next *incremental release plan* is selected to be executed.

The project manager must keep steady control of the project team to ensure that they deliver on timelines promised. The team members must not think that they have free rein to do what they want when they want because they might believe that a problem can always be corrected by an adaptive action. Only unforeseen circumstances or changes that have a direct impact on the success of a project may be used as excuses to log a change request or to perform an adaptive action.

It is important to realise that more than one *collaborative increment* can be in execution as the next *incremental release plan* can be selected once all dependencies on previously selected *collaborate increments* have been satisfied. The collaborative incremental boundaries are managed and controlled throughout the life cycle of the project.

After the successful completion and sign-off of each *collaborative increment*, a *full (system) implementation step* can be executed (only if necessary) to integrate the different *collaborative increments* in order to release them as a whole into the organisation. The system and processes are then handed over to the organisation in the *close-out and hand-over* phase, after which the system must be maintained and continually improved in the *post-project maintenance and continual improvement* phase.

Close-out and hand-over phase

Like all the PMMs, the hybrid APMM (ver. 2) also has a clear end. The *close-out and hand-over* phase is just as important as the other phases of the project. It is not an isolated phase, as it can be applied during any phase of the hybrid APMM (ver. 2). It can also be used to close a project down prematurely for whatever reason. During this phase, the project is formally closed off (even if the completion of the current project results in the kick-off of another project) by providing the client with all the signed-off deliverables, deliverables sets, collaborative increments and final implementation sign-off documentation (if relevant). The reason for obtaining formal closure is to ensure that the client or stakeholders do not later claim that certain expectations or functionalities were not provided. Furthermore, the final outcomes of the project must be measured against the vision, value and benefits that were committed to in the project schedule guideline and project charter.

The formal hand-over and training of the project or system is of high importance. In order to minimise documentation, full training can be provided to the users on utilising the system effectively together with a quick reference guide. The guide will refer to different important functionalities that are utilised by the user in his/her respective area of work. The development of a detailed manual is avoided because users normally become lost in voluminous manuals with too much information. An alternative is to create a help file using an index tool that would assist users in finding assistance quickly within their respective area of work. Only when users can utilise the system effectively without any assistance or training can project hand-over be viewed as being complete.

Many organisations require the documentation of detailed design specifications, which cannot be ignored. In a case such as this, it would be wise to consult with the client regarding the level of detail required in order to minimise the documentation workload.

Adapt, direct, monitor and control phase

This phase serves an integrating function, which takes place across all the different phases of the project. The different phases of the hybrid APMM (ver. 2) has the ability to adapt to their environment [and project nature](#). Projects of different sizes can be executed by replacing the collaborative development phase's *deliverables sets* with only a single deliverable as explained. The hybrid APMM (ver. 2) can easily and effectively adjust to [a changing project nature](#), and changing business and user requirements at a *collaborative incremental* level and iteration (deliverables set or deliverable) level during the *iterative refactoring cycle* by using adaptive actions instead of corrective actions. [Because the nature of a project changes, it is important to review the project's unique characteristics \(defined in the *pre-initiation* phase\) on a regular basis to determine](#)

whether an adaptive action is necessary and to ensure that the project's nature is observed and understood at all times.

The stakeholders, management, sponsor or project board (if appointed) supervises by putting certain project controls in place in order to make informed decisions based on facts not assumptions. They have the authority to provide *ad hoc* guidance as the project progresses. A culture of "respect for all" must be fostered together with the project manager, who must continually monitor, control and address any political or cultural issues that might arise as the project progresses. Human behaviour could also be studied, if there is sufficient time, to assist with the management of political issues and cultural behaviours. The stakeholders, management, sponsor or project board (if appointed) can assign project or organisational assurance to execute some of the assessing and reviewing functionality. They direct the project by considering exceptions as pointed out by the project manager, who acts as a communication channel by communicating all relevant project and progress information to programme or organisational management.

Continual monitoring, measuring and controlling of project-related activities is completed to ensure that the project team satisfy the deliverables of the project. A regular organisation communiqué will be distributed to the employees of the organisation to ensure that there is integrated communication across the organisation, and to ensure that employees and team members affected by project changes and the progress of a project are informed at all times. The team continually monitors and measures the progress of the project against the project schedule guideline in order to take the necessary adaptive actions if the project's scope, time, cost and quality constraints are threatened. The key objectives of this phase are to ensure that:

- authority has been provided to initiate the project;
- authority has been provided to sign-off the completed collaborative increments;
- the project remains viable, and that direction and control are provided throughout the duration of the project;
- programme and organisational management have an interface to the project;
- authority has been provided to formally close-out and hand-over the project;
- deliverables and functionalities are managed and reviewed to realise the post-project maintenance and continual improvement activities to be completed;
- the project's vision remains focused – what must be delivered by whom and by when– any deviation from the original value and benefit agreed upon in the project charter are monitored to avoid divergence from the focus, which might cause scope creep;
- the problems, cost, quality, issues and risks of the project are controlled;
- the *project schedule guideline* and structured backlog are reviewed regularly;

- the defined and accepted collaborative increments are delivered within the quality, time and cost constraints and tolerances to achieve the defined business benefits and value as stipulated in the project charter and project schedule guideline;
- the project work is monitored and controlled on a daily basis;
- the project follows an integrated change control process;
- the project scope, schedule guideline, quality and cost are controlled, monitored and verified regularly;
- the performance and progress of the project are tracked and reported on regularly;
- procurement is administrated;
- ensured that all issues are resolved quickly to minimise any negative impact on the project;
- the stakeholder map is kept up to date to ensure that the correct people are always informed and up to date on the progress, problems, and changes of the project; and
- environmental assessments are conducted to ensure that the project is up to date on the latest technologies and trends; this will ensure that the organisation remains competitive and gains from using newer and better technologies than its competitors.

In order to control and monitor a project effectively, it is important to report on the progress of a project in order to identify possible problems and problems that arose, and to identify adaptive actions before they become a threat to the successful delivery of a project. The escalation process of informing the stakeholders, management, sponsor or project board of identified risks, issues and changes to the project must also be formalised during this process.

All the meetings (stand-up, pre-incremental meeting, post-incremental meeting, feedback meetings, etc.) that will be conducted as the project progress should have an agenda of key outcomes and key decisions that must be made to ensure that the meeting has a specific purpose. It is the project manager's responsibility to ensure that the correct people are invited to a meeting and it will be the chair's responsibility to ensure that the discussions are focused on project-related topics. A principle that is emphasised by the hybrid APMM (ver. 2) is to show respect for everyone involved in the project, especially regarding political, human, religious and cultural behaviours and understanding.

Post-project maintenance and continual improvement phase

After the implementation, close-out and hand-over of a project, maintenance of the implemented solution should continue to ensure that the solution continues to bring value to the client organisation. In some cases, maintenance is done by the client organisation itself after hand-over.

There is always room for improvement to a solution, which is the reason that is important to monitor continually, whether improvement or an upgrade to the current implemented solution is necessary. As soon as users see what is possible, they always want more. For this reason, there are new releases of software and hardware products in the market to provide quicker and better solutions for clients. The hybrid APMM (ver. 2) life cycle is then re-initiated by the *vision and definition* phase if an improvement or upgrade is required. If pre-initiation planning is required, it can be done before the *vision and definition* phase is executed.

The basic structure of the hybrid APMM (ver. 0) remained the same; nothing was identified to be removed after the evaluation of the hybrid APMM (ver. 0 and ver. 1) in Chapter 6. The results of the case studies and survey only suggested changes and improvements that were incorporated to create the enhanced hybrid APMM (ver. 2).

7.4 Limitations

There are some aspects of the study that limited the gathering of information and generalisation of the results:

- The researcher was unable to interview more people because of the fast-paced and demanding environments in which the respondents work. In most cases, respondents were only able to grant sixty minutes for an interview because of their busy schedules and deadlines.
- Although the participants of both the large and small organisations stipulated that the hybrid APMM (ver. 0) could be applied in their environment (when the case studies were conducted), authorisation could not be received from senior management to test the hybrid APMM (ver. 0) in their project environment practically.
- Despite numerous follow-ups, phone calls and reminder e-mails, the researcher was unable to retrieve all the questionnaires sent to the sixty-five respondents. The researcher did however receive a considerable number of questionnaires (fifty-one).
- Suitable respondents were difficult to find at the beginning of the survey, as they needed to have a PMM or ASDM background to ensure that they were able to participate in the study meaningfully and thus yield high-quality data.
- The *critical project success criteria* in Section 5.2.3 might not have been complete, which may otherwise have resulted in other newly identified APSFs in the *agile project success criteria*.
- An automated hybrid APMM tool is not developed, which could have been used to guide project managers and other members to utilise the methodology correctly. This can be done as possible future work.

7.5 Future work

Like other PMMs, such as PMBOK and PRINCE2, the hybrid APMM (ver. 2) must still be reviewed and improved to make it more acceptable, implementable and applicable in practice. There is no “best breed” PMM or ASDM, and the aim of the hybrid APMM (ver. 2) is not to be a “best breed” PMM or ASDM, although it was found to be able to adapt to projects of different sizes and of different types in changing project and business environments.

The development of PMMs and ASDMs is a growing discipline in which project managers, software engineers and many other professionals aim to improve the way they manage and deliver projects because they realise that traditional methodologies have many flaws. Owing to this, there are many different ASDMs and versions of the PMBOK and PRINCE2 PMMs.

It is recommended that the hybrid APMM (ver. 2) be further evaluated in different sized organisations and in different project and business environments to investigate its effectiveness further and to improve it even further by addressing the concerns and flaws that arise in the process.

The recommendations for possible future work to enhance the hybrid APMM further could include:

- the evaluation and improvement of the hybrid APMM (ver. 2) by conducting more case studies and surveys in and outside South Africa in different project and business environments because every environment has its own characteristics that makes it unique in nature;
- the evaluation and improvement of the hybrid APMM (ver. 2) by applying it in practice;
- the improvement of the hybrid APMM (ver. 2) by combining it with other methodologies, frameworks, or standards such as CMMI, ISO9001, COBIT, or ITIL; and
- the evaluation of the degree of agility according to the agile characteristic features and values using the analytical framework, 4-DAT, as provided by Qumer and Henderson-Sellers (2008b:280).

The practical utilisation of a newly developed PMM or ASDM takes time, as most practitioners are cautious to apply a new methodology or a methodology that has not proven itself in practice. Good methodologies are practically and constantly tested and improved over time and must yield evidence that they aid the successful delivery of projects. The true value of the hybrid APMM (ver. 2) will only be realised after much practical application and improvement.

ANNEXURE A

Interview questions

1. Do you apply project management and system development principles in your daily job as a team member/project manager/programme manager/senior manager?
2. Do you think that there is a “best breed” project management or system development methodology that can be utilised for all types of projects?
3. Have you heard off and applied some of the aspects of PRINCE2/PMBOK as PMMs or other ASDMs in practice?
4. After studying the proposed hybrid APMM, do you think this hybrid APMM will assist project managers and team members in delivering IT projects successfully in a constantly changing project and business environment? (Yes/No) Why?
5. What is unique or new for you about the hybrid APMM provided?
6. After studying the hybrid APMM, in your opinion can it adapt to different types and sizes of projects? (Yes/No) How?
7. Based on your experience, what would you change in the hybrid APMM to make it more adaptable to delivering different types and different size projects in practice?
8. Do you believe in keeping a PMM/ASDM simple, minimising documentation, and focus on getting the actual work done? (Yes/No) Why?
9. What advantages or strengths did you identify in the proposed hybrid APMM?
10. What disadvantages or shortcomings did you observe in the hybrid APMM?
11. Do you think stakeholders and clients will be involved when the hybrid APMM is applied?
12. Do you think that maintenance and regular updates are important after the project has been delivered?
13. Do you think that change in a project will be adequately managed with the change-request form provided?
14. Do you think that regular sign-offs are important?
15. Do you think that the hybrid APMM will perform better than traditional PMMs or ASDMs?
16. What would you say are the most important drivers to ensure that a project is delivered successfully?
17. Do you think that planning can eliminate uncertainty?
18. Do you think it is a good idea to break a project down into increments and to develop each increment iteratively?
19. Do you think that lessons learnt must be logged as the project progresses, or only at the end of a project?
20. What would you say causes projects to fail?

ANNEXURE B

Change-request form

PROJECT INFORMATION						
Project description/name:			<i>Name of the project manager responsible for implementing the change</i>			
Programme/Project/Department/EPMO Manager:			<i>Name of the project for which the change is being requested</i>			
CHANGE INFORMATION						
Date of change request:		<i>Date on which the form was completed</i>				
Change rating/urgency:		<i>Urgency of the change (high, medium, or low)</i>				
Unique change number:		<i>Unique identifier for the requested change</i>				
Change requester:		<i>Name of person requesting the change</i>				
Change owner:		<i>Person responsible for ensuring that the change is implemented</i>				
<u>Change description:</u> <i>Describe the requested change in full.</i>						
<u>Change drivers:</u> <i>Describe/List the circumstances/events that caused the necessity/request for a change to the project.</i>						
<u>Change advantages:</u> <i>List the advantages of the change request.</i>			<u>Change disadvantages:</u> <i>List the disadvantages of the change request.</i>			
PROJECT IMPACT INFORMATION						
<i>Explain the positive and negative impact the change will have on the project scope, time, cost and quality constraints if the change is implemented or not implemented.</i>						
DOCUMENTATION						
<i>Reference and attach any supporting documentation.</i>						
APPROVAL DETAILS						
	Reviewed	Prepared	Supported	Accepted	Recommended	Approved
Signature						
Name						
Designation						
Date	2011/ /	2011/ /	2011/ /	2011/ /	2011/ /	2011/ /

ANNEXURE C

Questionnaire

QUESTIONNAIRE

i) ***IMPORTANT: Do not remove or add any lines whatsoever to this document**

ii) First read the Hybrid Agile Project Management Methodology.pdf before completing the questionnaire. Please note that this document may not be used or distributed without the prior approval of the author.

iii) In questions 2 and 4, the critical project success factors only explain and broaden the understanding of the various agile project success factors. Your answers to these questions must be according to a rating of 1 to 5 as follows:

- 1: Disagree completely
- 2: Disagree
- 3: Neutral
- 4: Agree
- 5: Agree completely

iv) The following terms are used in the questions. It is important that you understand these terms, as defined in this project:

PMM: A project management methodology prescribes the manner/way in which a project should be managed.

Agile: Explains the flexibility and adaptability of a methodology in delivering projects in a constantly changing project and business environment.

Critical project success factors: Factor or characteristic that is required for a project to be completed successfully.

APMM: An agile project management methodology prescribes the manner/way in which a project should be managed in a constantly changing project environment.

v) All information gathered by this questionnaire will be kept confidential.

vi) There are 5 questions, please complete all of them. If there are any problems or questions please contact Hannes at 083 xxx xxxx.

Question 1: Background information

Qualifications:

e.g. BSc in Computer Science

Position description:

e.g. Project Manager

Years of experience:

e.g. 3 years

Organisation size:

Select from list

Question 2:

Do you agree that the following agile project success factors are **relevant to agile projects?**

***Important:** Answer by placing a single “x” in the answer block for each of the 28 agile project success factors.

		Disagree completely <--> Agree completely				
Agile project success factors	Supporting critical project success factors	1	2	3	4	5
1. Management commitment	Executive/senior management ownership and support					
2. Organisational environment	- Adequate organisational change management - Stakeholders, users and team members must understand the benefits of successful project delivery for the business - Alignment of project and organisational objectives/vision - Impact analysis of the project on the organisation - Considering personal interests and ambition, but not placing these before the organisation's interests - Understanding existing business processes - Clearly defined and understood business drivers and objectives - Attempt to re-engineer business					
3. Team environment	Effective, aligned and cross-functional team organisational structure					
4. NEW: Project selection and prioritisation	Projects must be selected based on an accurate and sound business plan					

5. Team capability	• Trained, experienced and skilled project managers and team members • Good leadership • Need a strong project manager/leadership (A2, A3, B1, B2, B3) • Need the right people/resources on a team that is not short staffed with the right skills, allocated correctly and working together with the same vision – will result in a higher probability of project success (A1, A2, A3, B1, B2, B3)						
6. Client involvement	• User/client/stakeholder/sponsor involvement, participation, commitment, support, accountability and ownership • Consultation with stakeholders/users/clients • Considering stakeholders'/employees' recommendations, ideas and solutions • Keep stakeholders involved and provide regular feedback – this will ensure easier sign-offs (A1, A2, A3, B1, B2)						
7. NEW: Role definition	• Clearly defined roles, authority and responsibilities • Understanding the role of executive/senior management in the project • Stakeholder mapping (A1)						
8. NEW: Communication	• Constant and effective communication and feedback • Willingness to share information • Willingness to seek the advice of senior management • Integrated communication (A1, A3, B1, B2)						
9. NEW: Client expectations	• Managing unrealistic client and business expectations • Satisfied client, user, business and stakeholder expectations and requirements • Clearly defined, gathered and controlled user and business requirements and expectations • Stakeholders expect things to change immediately and they expect/force team members to deliver to unrealistic deadlines (A1, A2, A3, B1, B3) • Users do not know what they want (A1)						
10. NEW: Politics and culture	• Managing politics and culture • Human, behavioural, political and cultural factors are difficult to manage (A1, A3, B1)						
11. NEW: Collaborative teamwork	• A project requires a collaborative effort • Wellness and safety of team members are considered to be of high importance • Motivated team members • Quick and fair resolution of conflict • Team-member commitment • Self-controlled project team • Know your stakeholders and team (A3) • Project manager and team members must be on board to utilise and enforce a PMM (B1) • Team members feel that the project manager does not always understand (B3) • Performance management – people must be rewarded for doing a good job (A3, B1) • People talk too much and do too little (B3)						

	flow diagrams, and pair programming (This explanation was not a CPSF identified, it was merely used to explain to the respondents what is meant by this APSF.)								
19. Delivery strategy	• Use of prototyping • Do maintenance after implementation (B1, B2, B3) • Deliver quickly (A1) • Formally close-out, sign-off and hand-over a project (A1, B2, B3)								
20. NEW: Technical designs, practices and tools	• Sufficient technical design, practices and utilisation of planning tools • Performing a return on investment assessment • No overreliance on technology • Appropriate use and evaluation of, and experimentation with technology • Standard software infrastructure								
21. NEW: Testing	• High-quality testing								
22. Project nature	• Avoiding direction change in the middle of the project • Considering the growth and evolution of a project • Consider the nature of a project before applying a PMM/ASDM (A2, A3, B2, B3)								
23. Project type	• Different types of projects (hardware, software, infrastructure, architecture, etc.) (This explanation was not a CPSF identified, it was merely used to explain to the respondents what is meant by this APSF.)								
24. Project schedule	• Developing, tracking, adhering to, and correctly executing a detailed/robust project schedule and plan (verbal advice) • Acknowledgement that the project schedule and cost are inseparable • Adequate budgets and realistic timelines								
25. NEW: Resource management (availability and allocation)	• Availability of resources and technical competencies (skills, hardware, software, sufficient funds, etc.) • Sufficient resources (money, people, etc.) • Proactive resource allocation • Correct team and project manager selection and organisation • Need the right people/resources on a team that is not short staffed with the right skills, allocated correctly and working together with the same vision – will result in a higher probability of project success (A1, A2, A3, B1, B2, B3) • Ensure people are accountable for their tasks (A2)								
26. NEW: Project size	• Sufficient project sizing								
27. NEW: Project strategy	• Appropriate project strategy • Regular sign-offs are important, especially in a changing environment (A1, A2, A3, B2)								
28. NEWEST: Project environment	• Constantly fast-changing environment (dynamic) – change is the only constant (A1, A2, A3, B1, B2, B3) • Project environment is challenging (A1)								

Question 3

What other critical project success factors would you add for agile projects?

***Note:** If you do not want to add any, just leave the lines blank. You can add as many as you like.

Question 4

First read the Hybrid Agile Project Management Methodology.pdf.

Do you agree that the hybrid APM (ver. 1) **supports/addresses the following agile project success factors?**

		Disagree completely <-> Agree completely				
Agile project success factors	Supporting critical project success factors	1	2	3	4	5
1. Management commitment	Executive/senior management ownership and support					
2. Organisational environment	- Adequate organisational change management - Stakeholders, users and team members must understand the benefits of successful project delivery for the business - Alignment of project and organisational objectives/vision - Impact analysis of the project on the organisation - Considering personal interests and ambition, but not placing these before the organisation's interests - Understanding existing business processes - Clearly defined and understood business drivers and objectives - Attempt to re-engineer business					
3. Team environment	Effective, aligned and cross-functional team organisational structure					
4. NEW: Project selection and prioritisation	Projects must be selected based on an accurate and sound business plan					

5. Team capability	• Trained, experienced and skilled project managers and team members • Good leadership • Need a strong project manager/leadership (A2, A3, B1, B2, B3) • Need the right people/resources on a team that is not short staffed with the right skills, allocated correctly and working together with the same vision – will result in a higher probability of project success (A1, A2, A3, B1, B2, B3)						
6. Client involvement	• User/client/stakeholder/sponsor involvement, participation, commitment, support, accountability and ownership • Consultation with stakeholders/users/clients • Considering stakeholders'/employees' recommendations, ideas and solutions • Keep stakeholders involved and provide regular feedback – this will ensure easier sign-offs (A1, A2, A3, B1, B2)						
7. NEW: Role definition	• Clearly defined roles, authority and responsibilities • Understanding the role of executive/senior management in the project • Stakeholder mapping (A1)						
8. NEW: Communication	• Constant and effective communication and feedback • Willingness to share information • Willingness to seek the advice of senior management • Integrated communication (A1, A3, B1, B2)						
9. NEW: Client expectations	• Managing unrealistic client and business expectations • Satisfied client, user, business and stakeholder expectations and requirements • Clearly defined, gathered and controlled user and business requirements and expectations • Stakeholders expect things to change immediately and they expect/force team members to deliver to unrealistic deadlines (A1, A2, A3, B1, B3) • Users do not know what they want (A1)						
10. NEW: Politics and culture	• Managing politics and culture • Human, behavioural, political and cultural factors are difficult to manage (A1, A3, B1)						
11. NEW: Collaborative teamwork	• A project requires a collaborative effort • Wellness and safety of team members are considered to be of high importance • Motivated team members • Quick and fair resolution of conflict • Team-member commitment • Self-controlled project team • Know your stakeholders and team (A3) • Project manager and team members must be on board to utilise and enforce a PMM (B1) • Team members feel that the project manager does not always understand (B3) • Performance management – people must be rewarded for doing a good job (A3, B1) • People talk too much and do too little (B3)						

12. Project management process	• Adherence to high-quality standards and constraints for deliverables and the project • Defined checkpoints, gate reviews and structure • Correct application of project management theory • Correct selection of the correct PMM • Use and acceptance of the benefit of a defined, effective and standardised methodology • Understanding the advantages of project management • Processes and policies are defined with regard to project management best practices, approaches and concepts • Knowing that project management does not happen at executive level • Preventing scope creep • Reliable estimates • Adequate cash flow management and fund provision • Suitable and timeous sign-offs • Managing external project factors, such as vendor and regulatory issues • Need adequate funding/budget (A1) • Do benefit realisation regularly (A1, A2, A3, B1, B2, B3) • Keep PMM/ASDM simple, minimise documentation (keep important documents) and focus on completing the work (A1, A2, A3, B1, B2, B3) • Need a PMM that is flexible and agile – with the ability to adapt to change (A1, A2, A3, B1, B3) • Need integrated project management (A1, A3, B2)					
	13. Project definition process	• Clearly defined, understood and controlled business case, benefits, objectives, deliverables and project definition • Clear and unified vision/focus • Defining and minimising a robust project scope and controlling scope changes • Undefined/unclear scope (at beginning) – results in a great deal of rework (A1, A2)				
	14. NEW: Change management and control	• Effective change identification, control, management, processes, and procedures • Manage changing/unclear requirements – give stakeholders what they want, not what developers think they need (A2, B1, B2, B3) • Inform people of changes and the project managers must ensure that the changes are enforced (A1)				
	15. NEW: Monitoring and control	• Good quality control, management, processes, procedures, and acceptance criteria • Understandable monitoring and controlling processes and procedures to track project progress • Controlled project delivery • Standardised reporting and monitoring procedures • Resolving project problems identified as soon as possible • Maintaining and updating the WBS • Maintaining the project approval and sign-off processes • Require proper monitoring and control procedure (A1, A2)				
	16. NEW: Reporting and tracking	• Tracking and adhering to cost and budget estimates • Standardised reporting and monitoring procedures • Gathering lessons learnt as the project progresses • Log and implement lessons learnt as the project progresses to prevent repeating mistakes (A1, A2, A3, B1, B2, B3)				
17. NEW: Risk and issue management	• Regularly and promptly identifying, assessing, mitigating, resolving, reviewing, monitoring and controlling project risks and issues					
	18. Agile software techniques	• Techniques used during the execution of a project, such as entity relationship diagrams, decision tables, data-				

	flow diagrams, and pair programming (This explanation was not a CPSF identified, it was merely used to explain to the respondents what is meant by this APSF.)								
19. Delivery strategy	• Use of prototyping • Do maintenance after implementation (B1, B2, B3) • Deliver quickly (A1) • Formally close-out, sign-off and hand-over a project (A1, B2, B3) • Sufficient technical design, practices and utilisation of planning tools • Performing a return on investment assessment • No overreliance on technology • Appropriate use and evaluation of, and experimentation with technology • Standard software infrastructure • High-quality testing								
20. NEW: Technical designs, practices and tools									
21. NEW: Testing									
22. Project nature	• Avoiding direction change in the middle of the project • Considering the growth and evolution of a project • Consider the nature of a project before applying a PMM/ASDM (A2, A3, B2, B3)								
23. Project type	• Different types of projects (hardware, software, infrastructure, architecture, etc.) (This explanation was not a CPSF identified, it was merely used to explain to the respondents what is meant by this APSF.)								
24. Project schedule	• Developing, tracking, adhering to, and correctly executing a detailed/robust project schedule and plan (verbal advice) • Acknowledgement that the project schedule and cost are inseparable • Adequate budgets and realistic timelines								
25. NEW: Resource management (availability and allocation)	• Availability of resources and technical competencies (skills, hardware, software, sufficient funds, etc.) • Sufficient resources (money, people, etc.) • Proactive resource allocation • Correct team and project manager selection and organisation • Need the right people/resources on a team that is not short staffed with the right skills, allocated correctly and working together with the same vision – will result in a higher probability of project success (A1, A2, A3, B1, B2, B3) • Ensure people are accountable for their tasks (A2)								
26. NEW: Project size	• Sufficient project sizing								
27. NEW: Project strategy	• Appropriate project strategy • Regular sign-offs are important, especially in a changing environment (A1, A2, A3, B2)								
28. NEWEST: Project environment	• Constantly fast-changing environment (dynamic) – change is the only constant (A1, A2, A3, B1, B2, B3) • Project environment is challenging (A1)								

Question 5

After reading the hybrid APMM document, what improvements do you suggest to the methodology and why?

Important: If you have no improvements to add, simply state “nothing”.

REFERENCES

- ABRAHAMSSON, P., SALO, O., RONKAINEN, J. & WARSTA, J. 2002. Agile software development methods: Review and analysis.
http://scholar.google.co.za/scholar?q=Agile+software+development+methods:+Review+and+analysis&hl=en&as_sdt=0&as_vis=1&oi=scholar Date of access: 11 May 2011.
- AL-HAMDAN, Z. & ANTHONY, D. 2010. Deciding on a mixed-methods design in a doctoral study. *Nurse Researcher*, 18(1):45–56.
- ALIEV, R.A.O., FAZLOLLAHI, B. & ALIEV, R.R. 2004. Soft computing and its applications in Business and Economics. New York: Springer. 446 p.
- AMBLER, S. 2002. Agile development best dealt with in small groups. *Computing Canada*: 9, 26 Apr.
- AMBLER, S.W. & CONSTANTINE, L.L. 2002. The unified process transition and production phase: best practices in implementing the UT. Lawrence, Kansas: CMP Books. 309 p.
- ANDERSON, D.J. 2004. Agile management for software engineering: applying the theory of constraints for business results. New Jersey: Prentice-Hall. 313 p.
- ASHLEY, D. & ORENSTEIN, D.M. 2005. Sociological theory: classical statements. 6th ed. Boston, MA: Allyn and Bacon. 487 p.
- AUGUSTINE, S. 2005. Managing agile projects. Upper Saddle River, NJ: Prentice-Hall. 229 p.
- AUGUSTINE, S., PAYNE, B., SENCINDIVER, F. & WOODCOCK, S. 2005. Agile project management: steering from the edges. *Communications of the ACM*, 48(12):85–89, Dec.
- AVISON, D. & FITZGERALD, G. 2003. Information systems development: methodologies, techniques and tools. 3rd ed. London: McGraw-Hill. 592 p.
- AYDIN, M.N., HARMSSEN, F., VAN SLOOTEN, K. & STEGWEE, R.A. 2004. An information systems development method in use. *Turkish journal of electrical engineering*, 12(2):127–138. Available: Academic Search Premier. Date of access: 11 May 2011.
- BAINEY, K.R. 2004. Integrated IT project management: a model-centric approach. London: Artech House. 483 p.
- BAIRD, S. 2002. Sams teach yourself extreme programming in 24 hours. Indianapolis, IN: Sams. 454 p.
- BALLÉ, F. & BALLÉ, M. 2005. Lean development. *Business strategy review*, 16(3):17–22, Fall. Available: Business Source Premier. Date of access: 11 May 2011.
- BALTUS, B. 2001. Software quality: state of the art in management, testing and tools. New York: Springer. 288 p.
- BAZELEY, P. 2008. Mixed methods in management research. (In R. Thorpe & R. Holt, eds. The SAGE dictionary of qualitative management research. London: SAGE. p.133–136.)

- BBAN, A. 2008. Reconceptualization of the division between qualitative and quantitative research methods. *Cognition, brain, behaviour*, 12(4):337–343.
- BECK, K. 2000. *Extreme programming explained: embrace change*. Boston, MA: Addison Wesley. 174 p.
- BECK, K., BEEDLE, M., VAN BENNEKUM, A., COCKBURN, A., CUNNINGHAM, W., FOWLER, M., GRENNING, J., HIGHSMITH, J., HUNT, A., JEFFRIES, R., KERN, J., MARICK, B., MARTIN, R.C., MELLOR, S., SCHWABER, K., SUTHERLAND, J. & THOMAS, D. 2001. Manifesto for agile software development. <http://agilemanifesto.org/> Date of access: 1 Jul. 2009.
- BELL, D. 2009. Comparing the differences and complementary features of PRINCE2 and the PMI PMBOK guide. http://esi.hogwell.co.uk/resource_centre/white_papers/comparing%20the%20differences%20and%20complementary%20features%20of%20prince2%20and%20the%20pmi%20pmbok%20guide.pdf Date of access: 6 May 2011.
- BENBASAT, I., GOLDSTEIN, D.K. & MEAD, M. 1987. The case research strategy in studies of information systems. *MIS quarterly*, 11(3):369–386.
- BENJAMIN, C.O. 2009. *Engineering for business: theory and cases*. Maryland: University Press of America. 273 p.
- BENTLEY, C. 2005. *Practical PRINCE2*. Norwich: The Stationery Office. 315 p.
- BEYER, H. 2010. *User-centered agile methods*. San Rafael, CA: Morgan and Claypool. 70 p.
- BLOKDIJK, G. 2007. *Project management 100 success secrets*. Emereo Publishing. 120 p.
- BØDKER, K, KENSING, F. & SIMONSEN, J. 2004. *Participatory IT design: designing for business and workplace realities*. Cambridge, MA: MIT Press. 337 p.
- BOEHM, B. & TURNER, R. 2005. Management challenges to implement agile processes in traditional development organizations. *IEEE software*, 22(5):30–39.
- BOEHM, B.W. & TURNER, R. 2003. *Balancing agility and discipline: a guide for the perplexed*. Boston, MA: Addison-Wesley. 266 p.
- BOWERS, A.N., SANGWAN, R.S. & NEILL, C.J. 2007. Adoption of XP practices in the industry – a survey. *Software process: improvement and practice*, 12(3):283–294, May/Jun. Available: Computers and Applied Sciences Complete. Date of access: 11 May 2011.
- BOYER, K.K. & VERMA, R. 2010. *Operations and supply chain management for the 21st century*. Mason, OH: Cengage Learning. 534 p.
- BRINKKEMPER, S. 1996. Method engineering: engineering of information system development methods and tools. *Information and software technology*, 38:275–280.
- BROWN, J.D. 2001. *Using surveys in language problems*. Cambridge: Cambridge University Press. 319 p.

- BRUEGGE, B. & DUTOIT, A.H. 2010. Object-oriented software engineering: using UML, patterns, and Java. 3rd ed. Upper Saddle River, NJ: Prentice-Hall. 812 p.
- BURDETT, J. 2000. Changing channels: using the electronic meeting system to increase equity in decision making. *Information technology, learning and performance journal*, 18(2):3–12.
- BURKE, E.M. & COYNER, B.M. 2003. Java extreme programming cookbook. Sebastopol, CA: O'Reilly Media. 275 p.
- CHAN, F.K.Y. & THONG, J.Y.L. 2009. Acceptance of agile methodologies: a critical review and conceptual framework. *Decision support systems*, 46(4):803–814, Mar. Available: Science Direct. Date of access: 21 Nov. 2010.
- CHARVAT, J. 2003. Project management methodologies: selecting, implementing and supporting methodologies and processes for projects. Hoboken, NJ: Wiley. 264 p.
- CHEN, H.T. 2006. A theory-driven evaluation perspective on mixed-methods research. *Research in schools*, 13(1):75-83.
- CHEN, S, JAIN, L.C. & TAI, C. 2006. Computational economics: a perspective from computational intelligence. London/PA: IGI. 318 p.
- CHONG, J., PLUMMER, R., LEIFER, L., KLEMMER, S.R., OZGUR, E. and TOYE, G. 2005. Pair Programming: when and why it works. (*In* Romero, P., Good, J., Chaparro, E.A. and Bryant, S., eds. 17th Workshop of the Psychology of Programming Interest Group, Sussex University, p. 43-48, June)
- CHOW, T. & CHAO, D. 2008. A survey of critical success factors in agile software projects. *Journal of systems and software*, 81(6):961–971, Jun. Available: Science Direct. Date of access: 21 Nov. 2010.
- CHROMATIC. 2003. Extreme programming pocket guide. Sebastopol, CA: O'Reilly Media. 90 p.
- CLARKSON, I. 2010. PRINCE2 business benefits. http://www.best-management-practice.com/gempdf/PRINCE2_Business_Benefits_White_Paper_January2010.pdf Date of access: 6 May 2011.
- CLELAND, D.I. & GAREIS, R. 2006. The evolution of project management: global project management handbook. New York: McGraw-Hill. 575 p.
- COCKBURN, A. 2000. Just-in-time methodology construction. <http://alistair.cockburn.us/Just-in-time+methodology+construction> Date of access: 21 May 2011.
- COCKBURN, A. 2005. Crystal Clear: a human-powered methodology for small teams. Boston, MA: Addison-Wesley. 312 p.
- COHEN, D., LINDVALL, M. & COSTA, P. 2003. A DACS state of the art report: agile software development. <https://www.thedacs.com/techs/abstracts/abstract.php?dan=345473> Date of access: 17 May 2011.

- COHEN, J. 1988. Statistical power analysis for the behavioural sciences. Broadway Hillsdale, New Jersey: Routledge. 567 p.
- COHN, M. & SCHWABER, K. 2003. The need for agile project management. *Agile times*:1–4, Jan.
<http://www.mountangoatsoftware.com/system/article/file/14/ManagingAgileProjects.pdf?1267552461> Date of access: 6 May 2011.
- COMTE, A. 1896. The positive philosophy. Vol. 1. London: George Bell.
- CONBOY, K. & FITZGERALD, B. 2004. Towards a conceptual framework of agile methods. (In Zannier, C. et al., eds. 2004. Lecture notes in computer science: XP/Agile Universe. Vol. 3134. Berlin: Springer. p. 105–116.)
- CONTROL CHAOS. 2005. What is Scrum? <http://www.controlchaos.com> Date of access: 11 May 2011.
- CONWAY, K. 2000. Software project management: from concept to deployment. Scottsdale, AZ: Coriolis Group. 832 p.
- COOKE, J.L. 2010. Agile productivity unleashed: proven approaches for achieving real productivity gains in any organisation. Ely: IT Governance. 379 p.
- CRAWFORD, L. & BLACKBURN, S. 1996. Project manager competence: putting the PMBOK to work. 7 p. <http://www.projectperformance.com.au/downloads/96pd37.pdf> Date of access: 6 May 2011.
- CRESWELL, J.W. 1998. Qualitative inquiry and research design: choosing among five traditions. Thousand Oaks, CA: SAGE. 403 p.
- CRESWELL, J.W., FETTERS, M.D. & IVANKOVA, N.V. 2004. Designing a mixed methods study in primary care. *Annals of family medicine*, 2(1):7–12.
- CRISPIN, L. & HOUSE, T. 2003. Testing extreme programming. Boston, MA: Addison-Wesley. 306 p.
- DAHLKE, F. 2008. Eliminating waste in software projects: effective knowledge management by using web based collaboration technology. Hamburg: Diplomica. 82 p.
- DAUGHTREY, T. 2002. Fundamental concepts for the software quality engineer. Milwaukee, WI: ASQ Quality Press. 288 p.
- DE LUCA, J. 2005. Feature driven development overview.
<http://www.nebulon.com/articles/fdd/download/fddoverview.pdf> Date of access: 6 May 2011.
- DECARLO, D. 2004. Extreme project management: using leadership, principles and tools to deliver value in the face of volatility. San Francisco, CA: Wiley. 515 p.
- DENZIN, N.K. 1978. The research act: a theoretical introduction to sociological methods. 2nd ed. New York: McGraw-Hill. 370 p.

- DIAZ, A.E. 2006. Methodologies to implement ERP systems: are they PMBOK compliant? http://www.hunter-inc.com/pmi/ERP_PMBOK1.pdf Date of access: 6 May 2011.
- DILTHEY, W. 1989. The relationship of the human sciences to the natural sciences. (*In* Makkeel, R.A. & Frithjof, R., eds. Wilhelm Dilthey: Selected works. Vol. 1. Princeton, NJ: Princeton University Press. p. 66–71.)
- DINGSOYR, T., DYBA, T. & MOE, N.B. 2010. Agile software development: An introduction and overview. (*In* Dingsoyr, T., Dyba, T. & Moe, N.B., eds. Agile software development: current research and future directions Berlin, Heidelberg: Springer. p. 1-13).
- DOOM, C. 2010. An introduction to business information management. Brussels: Asp/Vubpress/Upa. 170 p.
- DOSS, G.M., WALLACE, M. & WEBBER, L. 2004. IS project management handbook. New York: Aspen. 562 p.
- DSDM Consortium. 2005. The history of the DSDM consortium. <http://www.dsdm.org/en/about/history.asp> Date of access: 29 Mar. 2005.
- DU PLESSIS, P. & MAJAM, T. 2010. Mixed method research a new paradigm? *Journal of public administration*, 45(3):456–472, Sept.
- DUFFY, T. M. & JONASSEN, D. H. 1991. New implications for instructional technology? *Educational technology*, 31(3): 7 – 12.
- DUNNING, H, WILLIAMS, A, ABONYI, S. & CROOKS, V. 2007. A mixed method approach to quality of life research: a case study approach. *Social indicators research*, 85(1):145–158, May.
- DURKHEIM, E. 1985. Sociology and the social sciences. (*In* Thomson, K., ed. Readings from Emile Durkheim. London: Routledge. p. 21–27.)
- EISENHARDT, K.M. 1989. Building theories from case study research. *Academy of management review*, 14:532–550.
- ELLIS, S.M. & STEYN, H.S. 2003. Practical significance (effect sizes) versus or in combination with statistical significance (p-values). *Management dynamics*, 12(4):51–53.
- FEATHER, S.R. 1999. The impact of group support systems on collaborative learning groups' stages of development. *Information technology, learning and performance journal*, 17(2):23–34.
- FINK, A. 2003. The survey kit: the survey handbook. Thousand Oaks, CA: SAGE. 184 p.
- FITSILIS, P. 2008. Comparing PMBOK and agile software development process. (*In* Sobh, T., ed. Advances in computer and information sciences and engineering. Berlin: Springer. p. 378–383.)
- FLETCHER, K. 2006. Partnerships in social care: a handbook for developing effective services. Philadelphia, PA: Jessica Kingsley. 139 p.
- FOWLER, M. 2001. Put your process on a diet. www.sdmagazine.com/documents/s=737/sdm0012a/0012a.htm Date of access: 19 April 2005.

- FOWLER, M. 2005. The new methodology. <http://martinfowler.com/articles/newMethodology.html>
Date of access: 21 May 2011.
- FURLONG, I.S. & AL-KARAGHOULI, W. 2009. The evaluation and the effectiveness of project management in transformational e-government projects. European and Mediterranean Conference on Information Systems (EMCIS2009). Crowne Plaza Hotel. 13–14 July 2009. 12 p.
- GIDDENS, A. 1974. Positivism and sociology. London: Heinemann. 244 p.
- GIGGINGS, L.S. 2006. Mixed methods research: positivism dressed in drag? *Journal of research in nursing*, 11(3):195–203.
- GLOBERSON, S. & ZWIKAEI, O. 2002. The impact of the project manager on project management planning processes. *Project management journal*, 33(3):58–64, Sept.
- GOOD, J.M. 2003. A pragmatic approach to the implementation of agile software development methodologies in plan-driven organisations. Canterbury: Lincoln University. (Dissertation – Hons. Degree.) 84 p.
- GOODPASTURE, J.C. 2010. Project management the agile way: making it work in the enterprise. Fort Lauderdale, FL: J. Ross. 350 p.
- GOTTSCHALK, P. 2006. E-business strategy, sourcing and governance. Hershey, PA: IGI. 351 p.
- GRAHAM, N. 2010. PRINCE2 for dummies. West Sussex: Wiley. 392 p.
- GREENE, J.C., CARACELLI, V.J. & GRAHAM, W.F. 1989. Toward a conceptual framework for mixed-methods evaluation designs. *Educational evaluation and policy analysis*, 11(3):255–274.
- GUPTA, B.L. 2007. Governance and management of technical institutions. New Delhi: Concept. 570 p.
- HALLOWS, J. 2005. Information systems project management: how to deliver function and value in information technology projects. New York: AMACOM. 286 p.
- HARDY, C.J., THOMPSON, J.B. & EDWARDS, H.M. 1995. The use, limitations and customization of structured systems development in the United Kingdom. *Information and systems technology*, 37(9):467–477.
- HARRISON, F.L. & LOCK, D. 2004. Advanced project management: a structured approach. Aldershot: Gower. 315 p.
- HASS, K.B. 2007. The blending of traditional and agile project management. *PM world today*, IX(V):1–8, May.
- HASS, K.B. 2009. Managing complex projects: a new model. Vienna, VA: Management Concepts. 298 p.
- HAUGAN, G.T. 2008. Work breakdown structures for projects, programs and enterprises. management concepts. Vienna, VA: Management Concepts. 382 p.

- HAUGAN, G.T. 2011. Project management fundamentals: key concepts and methodology. Vienna, VA: Management Concepts. 367 p.
- HAYES, I.S. 2003. Just enough wireless computing. Upper Saddle River, NJ: Prentice-Hall. 416 p.
- HEDEMAN, B. & VIS VAN HEEMST, G. 2006. Programme management based on MSP: an introduction. Zaltbommel: Van Haren. 252 p.
- HEDEMAN, B., VIS VAN HEEMST, G. & FREDRIKSZ, H. 2010. Project management based on PRINCE2 2009 Edition. Zaltbommel: Van Haren. 244 p.
- HEVNER, A.R., MARCH, S.T., PARK, J. and RAM, S. 2004. Design science in information systems research. *MIS Quarterly*, 28(1):75-105, Mar.
- HEVNER, A. R. 2007. A three cycle view of design science research. *Scandinavian Journal of Information Systems*, 19(2):87-92.
- HEWSON, C. 2006. Mixed methods research. (In Jupp, V., ed. *The SAGE dictionary of social research methods*. London: Sage. p. 179–180.)
- HIGHSMITH, J. 2000a. Retiring lifecycle dinosaurs. *Software testing and quality engineering*:22–28, Jul./Aug. <http://www.adaptivesd.com/articles/Dinosaurs.pdf> Date of access: 6 May 2011.
- HIGHSMITH, J. 2000b. Adaptive software development: a collaborative approach to managing complex systems. New York: Dorset House. 358 p.
- HIGHSMITH, J. 2002a. What is agile software development? *Crosstalk*:4–9, Oct. http://www.adaptivesd.com/articles/cross_oct02.pdf Date of access: 6 May 2011.
- HIGHSMITH, J. 2002b. Extreme programming: agile project management advisory white paper service. <http://rockfish.cs.unc.edu/COMP290-agile/xp-highsmith.pdf> Date of access: 6 May 2011.
- HIGHSMITH, J. 2002c. Agile software development ecosystems. Boston, MA: Addison-Wesley. 448 p.
- HIGHSMITH, J. 2010. Agile project management: creating innovative products. 2nd ed. Boston, MA: Addison-Wesley. 432 p.
- HIGHSMITH, J. & COCKBURN, A. 2001a. Agile software development: the business of innovation. *IEEE computer*:120–122, Sept.
- HIGHTECH DIMENSIONS. 2002. Why do IT (Information Technology) projects fail? Reston, VA: Hitech Dimensions. 32 p.
- HISLOP, G.W., LUTZ, M.J., NAVEDA, J.F., MCCracken, W.M., MEAD, N.R. & WILLIAMS, L.A. 2002. Integrating agile practices into software engineering courses. *Computer science education*, 12(3):169–185.
- HOLCOMBE, M. & HOLCOMBE, W.M.L. 2008. Running an agile software development project. Hoboken, NJ: Wiley. 312 p.

- HOLDSWORTH, B. & MARTIN, G.R., *ed.* 1993. Digital systems reference book. Oxford: Butterworth-Heinemann. 1050 p.
- HOLMSTROM, H., FITZGERALD, B., AGERFALK, P.J. & CONCHUIR, E.O. 2006. Agile practices reduce distance in global software development. *Information systems management*, 23(3):7–18, Summer. Available: Business Source Premier. Date of access: 17 May 2011.
- HOWCROFT, D. & TRAUTH, E.M. 2004. The choice of critical information systems research. *IFIP International federation for information processing*, 143(1):195–212.
- HUGHES, B. & COTTERELL, M. 2002. Software project management. 3rd ed. London: McGraw-Hill. 358 p.
- HUGHES, J.A. 1990. The philosophy of social research. New York: Longman. 169 p.
- HUIJBERS, R., LEMMENS, F., SENDERS, B., SIMONS, S., SPAAN, B., VAN TILBURG, P. & VOSSSEN. 2004. Software project management: methodologies and techniques. (Software Engineering Project presented as part of a study at the Technische Universiteit Eindhoven on 17 Sept. 2004.) Eindhoven. 37 p. (Unpublished.)
- HUISMAN, H.M. & IIVARI, J. 2006. Deployment of systems development methodologies: Perceptual congruence between IS managers and system developers. *Information and management*, 43:29–49.
- HUNT, J. 2006. Agile software construction. Wiltshire: Springer. 254 p.
- IIVARI, J. & MAANSAARI, J. 1998. The usage of systems development methods: are we stuck to old practices? *Information and software technology*, 40:501–510, Jun.
- IIVARI, J., HIRSCHHEIM, R. & KLEIN, H.K. 1999. Beyond methodologies: keeping up with information systems development approaches through dynamic classification. (*In* Proceedings of the 32nd Hawaii International Conference on System Sciences, ISJ:6(1)3–24.)
- JEFFRIES, R. 2001. What is Extreme Programming? <http://xprogramming.com/what-is-extreme-programming/> Date of access: 21 May 2011.
- JICK, T.D. 1979. Mixing qualitative and quantitative methods: triangulation in action. *Administrative science quarterly*, 24:602–611.
- JOHNSON, B. & TURNER, L.A. 2003. Data collection strategies in mixed methods research. (*In* Tashakori, A. & Teddlie, C., *eds.* Handbook of mixed methods in social and behavioural research. Thousand Oaks, CA: SAGE. p. 297–319.)
- JOHNSON, R.B. & ONWUEGBUZIE, A.J. 2004. Mixed methods research: a research paradigm whose time has come. *Educational researcher*, 33(7):14–26, Oct.
- JOHNSON, R.B., ONWUEGBUZIE, A.J. & TURNER, L.A. 2007. Toward a definition of mixed methods research. *Journal of mixed methods research*, 1(2):112–133.
- JONES, R. 2007. Project management survival: a practical guide to leading, managing and delivering challenging projects. Philadelphia, PA: Kogan Page. 242 p.

- KAMMERMEIER, M. 2009. Agile project management in IT development projects with a focus on team performance. Norderstedt: GRIN. 52 p.
- KARLESKY, M. & VAN DER VOORD, M. 2008. Agile project management (or, burning your Gantt charts). <http://www.atomicobject.com/files/EmbeddedAgilePMPaper.pdf> Date of access: 6 May 2011.
- KASSE, T. 2004. Practical insight into CMMI. Norwood, MA: Artech House. 281 p.
- KELLY, A. 2008. Reflections on PRINCE2 from an agile perspective. 41 p.
<http://www.allankelly.net/static/writing/webonly/ReflectionsOnPRINCE2.pdf>. Date of access: 6 May 2011.
- KERZNER, H. 2004. Advanced project management: best practices on implementation. Hoboken, NJ: Wiley. 847 p.
- KERZNER, H. 2010. Project management best practices: achieving global excellence. Hoboken, NJ: Wiley. 704 p.
- KIM, C., PETERSON, D., CHIN, J. & BARRIER, T. 2001. Critical strategies for information systems development projects: perceptions of developers in Korea. (*In* Tan, F.B., ed. 2002. Global perspective of information technology management. Hershey, PA: IGI. p. 179-189)
- KLEIN, H.K. & MYERS, M.D. 1999. A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS quarterly*, 23(1):67–94.
- KLOPPENBORG, T.J. 2009. Contemporary project management. Mason, OH: South-Western Cengage Learning. 480 p.
- KNUTSON, J. 2001. Project management for business professionals: a comprehensive guide. New York: Wiley. 596 p.
- KOSKELA, L. & HOWELL, G. 2001. Reforming project management: the role of planning, execution and controlling.
http://usir.salford.ac.uk/9384/1/2001_Reforming_Project_Management_the_role_of_planning_execution_and_controlling.pdf Date of access: 6 May 2011.
- LARMAN, C. 2004. Agile and iterative development: a manager's guide. Boston, MA: Addison-Wesley. 342 p.
- LARMAN, C. & VODDE, B. 2009. Lean primer. 46 p.
http://www.leanprimer.com/downloads/lean_primer.pdf Date of access: 6 May 2011.
- LEECH, N.L. & ONWUEGBUZIE, A.J. 2009. A typology of mixed methods research design. *Quality and quantity: international journal of methodology*, 43:265–275.
- LENZ, G. & MOELLER, T. 2004. NET: a complete development cycle. Boston, MA: Addison-Wesley. 553 p.
- LIENTZ, B.P. & REA, K.P. 2002. International project management. San Diego, CA: Academic Press. 277 p.

- LINDSTROM, L. & JEFFRIES, R. 2004. Extreme programming and agile software development methodologies. *Information systems management*:41–52, Summer.
- LINDVALL, M., BASILI, V., BOEHM, B., COSTA, P., KATHLEEN, D., SHULL, F., TESORIERO, R., WILLIAMS, L.A. & ZELKOWITZ, M.V. 2002. Empirical findings in agile methods. (*In* Wells, D. & Williams, L., eds. *Lecture notes in computer science: Extreme Programming and agile methods — XP/Agile Universe 2002*. Vol. 2418. Berlin: Springer. p. 81–92.)
- LIU, L. & ROUSSEV, B. 2006. *Management of the object-oriented development process*. Hershey, PA: Idea Group. 372 p.
- LIVERMORE, J.A. 2008. Factors that significantly impact the implementation of an agile software development methodology. *Journal of software*, 3(4):31–36, Apr. Available: Computers and Applied Science Complete. Date of access: 17 May 2011.
- LOCK, D. 2007. *Project management*. Aldershot: Gower. 520 p.
- LUBBE, S. 2003. The development of a case study methodology in the information technology field in South Africa: a step-by-step approach. *South African journal of information management*, 5(4):Dec.
- MACK, L. 2010. The philosophical underpinnings of educational research. *Polyglossia*, 19:5–11. Oct.
- MARCH, S. & SMITH, G. 1995. Design and natural science research on information technology. *Decision support systems*, 15:251–266.
- MCMASTER, T., WASTELL, D., FERNELY, E. & DEGROSS, J.I., eds. 2007. *Organizational dynamics of technology-based innovation: diversifying the research agenda*. New York: Springer. 534 p.
- MELTON, T. 2007. *Project management toolkit: the basics for project success*. Oxford: Butterworth-Heinemann. 286 p.
- MENDONCA, J. 2002. The case for a less methodical methodology: lean, light, extreme, adaptive, agile and appropriate software development. (*In* Khosrow-Pour, M., ed. *Issues and trends of information technology management in contemporary organisations*. Vol. 1. Hershey, PA.: Idea Group. p. 503–505.)
- MINGERS, J. 2001. Combining IS research methods: towards a pluralist methodology. *Information systems research*, 12(3):240–259, Sept.
- MOLINA-ARORIN, J.F. 2010. The use and added value of mixed methods in management research. *Journal of mixed methods research*, 5(1):6–24, Nov.
- MOORE, S. 2010. *Strategic project portfolio management: enabling a productive organisation*. Hoboken, NJ: Wiley. 176 p.
- MORRIS, P.W.G. 1988. Managing project interfaces: key points for project success (*In* Cleland, D.I. & King, W.R., eds. *Project management handbook*. New York: Van Nostrand Reinold. p 16–55.)

- MÜLLER, R. 2009. Project governance. Aldershot: Gower. 105 p.
- MUFFATTO, M. 2006. Open source: a multidisciplinary approach. London: Imperial College Press. 245 p.
- MURRAY, A. 2009. Managing and directing successful projects with PRINCE2. http://www.best-management-practice.com/gempdf/PRINCE2_2009_Overview_Brochure_June2009.pdf Date of access: 6 May 2011.
- MURRAY, A. 2007. Everything you wanted to know about PRINCE2 in less than one thousand words. http://www.best-management-practice.com/gempdf/PRINCE2_White_Paper_v3.pdf Date of access: 6 May 2011.
- MYER, T. 2008. Software development the agile way. *Baselinemag*, 81:56–57, Feb. Available: Business Source Premier. Date of access: 17 May 2011.
- NAGL, M. 1996. Building tightly integrated software development environments: the IPSEN approach. Berlin, Heidelberg: Springer. 709 p.
- NELSON, C. & NELSON C.E. 2006. Managing quality in architecture: a handbook for creators of the built environment. Oxford: Architectural Press. 318 p.
- NERUR, S., MAHAPATRA, R. & MANGALARA, G. 2005. Challenges of migrating to agile methodologies. *Communications of the ACM*, 48(5):73–78, May. Available: Academic Search Premier. Date of access: 17 May 2011.
- NGULUBE, P. 2010. Mapping mixed methods research in library and information science journals in sub-Saharan Africa 2004–2008. *International information and library review*, 42:252–261.
- NORTON, D. 2005. Lean software development overview. <http://codebetter.com/darrellnorton/2005/02/02/lean-software-development-overview/> Date of access: 17 May 2011.
- NUNAMAKER, J.F., CHEN, M. and PURDIN, T.D.M. 1990-1991. System development in information systems research. *Journal of Management Information Systems*, 7(3):89-106, Winter.
- Ó CONCHÚIR, D. 2010. Overview of the PMBOK guide: shortcuts for PMP certification. Berlin: Springer. 213 p.
- OATES, B.J. 2006. Researching information systems and computing. Thousand Oaks, CA: SAGE. 341 p.
- ODUM, J.N. 2004. Sterile product facility design and project management. 2nd ed. Boca Raton, FL: CRC Press. 373 p.
- OFFICE OF GOVERNMENT COMMERCE. 2002. Tailoring PRINCE2. Norwich: The Stationery Office. 138 p.
- OFFICE OF GOVERNMENT COMMERCE. 2009a. Managing successful projects with PRINCE2. Norwich: The Stationery Office. 327 p.

- OFFICE OF GOVERNMENT COMMERCE. 2009b. An introduction to PRINCE2: managing and directing successful projects. Norwich: The Stationery Office. 123 p.
- OFFICE OF GOVERNMENT COMMERCE. 2010. Delivering IT services using ITIL. Norwich: The Stationery Office. 210 p.
- OGC **see** OFFICE OF GOVERNMENT COMMERCE.
- OXFORD UNIVERSITY PRESS. 2011. Positivism.
<http://www.oxforddictionaries.com/definition/positivism?view=uk> Date of access: 2 May 2011.
- PALMER, S.R. & FELSING, J.M. 2002. A practical guide to feature-driven development. Upper Saddle River, NJ: Prentice-Hall. 271 p.
- PALVIA, P. & NOSEK, J.T. 1993. A field examination of system lifecycle techniques and methodologies. *Information and management*, 25:73–84.
- PEARMAN, G. & GOODWILL, J. 2006. Pro .NET 2.0 extreme programming. Berkeley, CA: Apress. 319 p.
- PEFFERS, K., TUUNANEN, T., ROTHENBERGER, M.A. and CHATTERJEE, S. 2007-2008. A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3):45-78, Winter.
- PENCE, R. 2007. Score with agile software development. *Systems iNews*, 333:34–38, Jul. Available: Computers and Applied Sciences Complete. Date of access: 17 May 2011.
- PERRY, M.P. 2009. Business driven PMO setup: practical insights, techniques and case examples for ensuring success. Fort Lauderdale, FL: J. Ross. 494 p.
- PETERSEN, K. & WOHLIN, C. 2009. A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. *Journal of systems and software*:1479–1490. Available: Science Direct. Date of access: 21 Nov. 2010.
- PHILLIPS, D. 2004. E-software project manager's handbook: principles that work at work. Hoboken, NJ: Wiley–IEEE. 485 p.
- PHIN, L.N., ZAMANI, A.Z. & KHENG, K.S. 2005. Multimedia super corridor integration architecture framework for large scale government applications. (Paper presented at the PMI Global Congress at the Swissotel the Stamford on 21–23 Feb 2005.) Singapore. 8 p.
http://www.asiaictpm.org/whitepaper/1237576697_asiaictpm_pmbok_final_paper.pdf Date of access: 3 Jun. 2011.
- PM4DEV. 2007. Fundamentals of project management. Norcross, GA: PM4DEV. 132 p.
- PMBOK GUIDE. 2008. A guide to the project management body of knowledge. 4th ed. Newtown Square, PA: PMI Publishers. 467 p.
- POPPENDIECK, M. 2003. Lean software development. *C++ magazine: methodology issue*, Fall.
http://www.poppendieck.com/pdfs/Lean_Software_Development.pdf Date of access: 4 Apr. 2006.

- POPPENDIECK, M. & POPPENDIECK, T. 2003. Lean software development: and agile toolkit. Upper Saddle River, NJ: Addison-Wesley. 203 p.
- POPPENDIECK, M. 2002. Principles of lean thinking. 7 p.
<http://www.hiirc.org.nz/page/17459/principles-of-lean-thinking-paper-by-mary/> Date of access: 17 May 2011.
- PORTMAN, H. 2009. PRINCE2 in practice, a practical approach to creating project management documents. Zaltbommel: Van Haren. 122 p.
- PROJECT PLUS. 2008. Providing clarity on PMBOK and PRINCE2.
<http://www.projectplusgroup.co.nz/page51001.html> Date of access: 17 Jun. 2010.
- PURI, C.P. 2009. Agile management: feature driven development. New Delhi: Global India. 310 p.
- QUMER, A. & HENDERSON-SELLERS, B. 2008a. A framework to support the evaluation, adoption and improvement of agile methods in practice. *Journal of systems and software*, 81:1899–1919. Available: Science Direct. Date of access: 21 Nov. 2010.
- QUMER, A. & HENDERSON-SELLERS, B. 2008b. An evaluation of the degree of agility in six agile methods and its applicability for method engineering. *Information and software technology*, 50:280–295. Available: Science Direct. Date of access: 21 Nov. 2010.
- READ, D. & PROPERJOHN, G. Going agile – a case study.
<http://www.ss.com.au/articles/Going%20Agile%20-%20A%20Case%20Study.pdf> Date of access: 6 May 2011.
- REIFER, D.J. 2002. How good are agile methods? *IEEE software*:16–18, Jul./Aug.
- RIBEIRO, J.M. 2009. Procurement of goods, works and services in development projects: with an overview of project. Montréal: Presses Internationales Polytechnique. 118 p.
- RICHARDS, K. 2007. Agile project management: running PRINCE2 projects with DSDM Atern. Norwich: The Stationery Office. 98 p.
- RICO, D.F. 2010. Lean and agile project management: for large programs and projects. *Lecture notes in business information processing*, 65(1):37–43.
- RICO, D.F., SAYANI, H.H. & SONE, S. 2009. The business value of agile software methods: maximizing ROI with just-in-time processes and documentation. Fort Lauderdale, FL: J. Ross. 214 p.
- RIEZEBOS, J. KLINGENBERG, W. & HICKS, C. 2009. Lean production and information technology: connection or contradiction? *Computers in industry*, 60(4):237–247, May. Available: Business Source Premier. Date of access: 17 May 2011.
- ROCCO, T.S., BLIS, L.A., GALLAGHER, S. & PEREZ-PRADO, A. 2003. Taking the next step: mixed methods research in organizational systems. *Information technology, learning and performance journal*, 21(1):19–29, Spring.

- ROSENBERG, D., STEPHENS, M. & COLLINS-COPE, M. 2005. Agile development with ICONIX process: people, process and pragmatism. Berkeley, CA: Apress. 261 p.
- RUNES, D.D. 1962. Dictionary of philosophy. 15th ed. Paterson, NJ: Littlefield, Adams and Co.
- RYCHLY, M. & TICHA, P. 2008. A tool for supporting feature-driven development. p. 196–207. (In Meyer, B., Nawrocki, J.R. & Walter, B., eds. Balancing agility and formalism in software engineering: Second IFIP TC 2 Central and East European Conference on Software Engineering Techniques, CEE-SET 2007, Poznań, Poland, Oct. 10–12, 2007: revised selected papers. Berlin: Springer. p. 196–207.)
- SAEKI, M. 1998. A meta-model for method integration. *Information and software technology*, 39:25–932.
- SAHAKIAN, W.S. 1968. History of philosophy: from the earliest times to the present. New York: Barnes and Noble. 366 p.
- SALADIS, F.P. & KERZNER, H. 2009. Bringing the PMBOK guide to life: a companion for the practicing project manager. Hoboken, NJ: Wiley. 288 p.
- 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. *IET software*, 2(1):58–64, Feb. Available: Academic Search Premier. Date of access: 17 May 2011.
- SCHACH, S.R. 1997. Software engineering with Java. New York: Irwin. 618 p.
- SCHUH, P. 2004. Integrating agile development in the real world. Hingham, MA: Charles River Media, Inc. 346 p.
- SCHWABER, K. & BEEDLE, M. 2002. Agile software development with Scrum. Upper Saddle River, NJ: Prentice-Hall. 158 p.
- SCHWALBE, K. 2010. Information technology project management. 6th ed. Boston, MA: Cengage Learning. 490 p.
- SEAMAN, C.B. 1999. Qualitative methods in empirical studies of software engineering. *IEEE transactions on software engineering*, 25(4):557–572.
- SELIG, G.J. 2008. Implementing IT governance: a practical guide to global best practices in IT management. Zaltbommel: Van Haren. 298 p.
- SHELLY, G.B. & ROSENBLATT, H.J. 2010. Systems analysis and design. Boston, MA: Cengage Learning. 731 p.
- SHERRER, A.J. 2009. Project management road trip for the project manager professional. J. Alex Sherrer. 571 p.
- SIEGELAUB, J. M. 2006. How PRINCE2 can complement PMBOK and your PMP. <http://www.corpedgroup.com/resources/pm/HowPrince2CanComplement.asp> Date of access: 14 November 2011.

- SINGH, K. 2007. Quantitative social research methods. New Delhi: SAGE. 431 p.
- SLIGER, M. & BRODERICK, S. 2008. The software project manager's bridge to agility. Boston, MA: Addison-Wesley. 353 p.
- SLIGER, M. 2006. A project manager's survival guide to going agile.
http://www.rallydev.com/documents/rally_survival_guide.pdf Date of access: 6 May 2011.
- SNYDER STACKPOLE, C. 2008. PMBOK guide – fourth edition changes.
<http://allpm.com/modules.php?op=modload&name=News&file=article&sid=2060> Date of access: 6 May 2011.
- SNYDER STACKPOLE, C. 2010. A user's manual to the PMBOK guide. Hoboken, NJ: Wiley. 240 p.
- SPINNER, M. 1997. Project management: principles and practices. London: Prentice-Hall. 306 p.
- STAHL, B.C. 2008. The ethical nature of critical research in information systems. *Information systems journal*, 18(2):137–163, Mar.
- STAMELOS, I.G. & SFETSOS, P. 2007. Agile software development quality assurance. Hershey, PA: IGI. 246 p.
- STANDISH GROUP INTERNATIONAL. 2001. Extreme chaos.
http://standishgroup.com/sample_research/extreme_chaos.pdf Date of access: 17 May 2011.
- STAPLETON, J. 1997. DSDM: dynamic systems development method: the method in practice. Harlow: Addison-Wesley. 163 p.
- STEINDL, C. 2004. Lean software development.
<http://agilealliance.org/articles/steindlchristophleans/file> Date of access: 3 Apr. 2006.
- STOBER, T. & HANSMANN, U. 2010. Agile software development: best practices for large software development projects. Berlin Heidelberg: Springer. 179p.
- STRAUSS, A.L. and CORBIN, J. 1990. Basics of qualitative research: grounded theory procedures and techniques. London: Sage. 270p.
- TAFT, D.K. 2005. Microsoft commits to agility. *Eweek.com*, 22(45):15–16, Nov. Available: Academic Search Premier. Date of access: 17 May 2011.
- TASHAKKORI, A. & CRESWELL, J.W. 2007. Editorial: The new era of mixed methods. *Journal of mixed methods research*, 1(1):3–7.
- TEDDLIE, C. & TASHAKKORI, A. 2003. Major issues and controversies in the use of mixed methods in the social and behavioural sciences. (In Tashakori, A. & Teddlie, C., eds. Handbook of mixed methods in social and behavioural research. Thousand Oaks, CA: SAGE. p. 3–50.)
- THAYER, R.H., ed. 1997. Software engineering project management. Los Alamitos, CA: IEEE Computer Society. 531 p.
- THEJENDRA, B.S. 2008. Disaster recovery and business continuity. Ely: IT Governance. 304 p.

- THURMOND, V.A. 2001. The point of triangulation. *Journal of Nursing Scholarship*, 33(3):253-258.
- TIKU, G.L. 2002. A manual on project management. New Delhi: Atlantic. 144 p.
- TOLFO, C. & WAZLAWICK, R.S. 2008. The influence of organizational culture on the adoption of extreme programming. *Journal of systems and software*, 81(11):1955–1967, Nov. Available: Business Source Premier. Date of access: 17 May 2011.
- TONCHIA, S. & COZZI, F. 2008. Industrial project management: planning, design and construction. Berlin Heidelberg: Springer. 229 p.
- TOOLEY, M. & DINGLE, L. 2002. BTEC national engineering. Oxford: Newnes. 496 p.
- VAISHNAVI, V. & KEUCHLER, B. 2004. Design research in information systems, AISWorldnNet. <http://desrist.org/design-research-in-information-systems/> Date of access: 21 May 2011.
- VAN BON, J. & VERHEIJEN, T. 2006. Frameworks for IT management. Zaltbommel: Van Haren. 227 p.
- VERZUH, E. 2008. The fast forward MBA in project management. Hoboken, NJ: Wiley. 462 p.
- VIRINE, L. & TRUMPER, M. 2008. Project decisions: the art and science. Vienna, VA: Management Concepts. 344 p.
- WALSHAM, G. 1995. Interpretive case studies in IS research: nature and method. *European journal of information systems*, 4:74–81.
- WANGENHEIM, C.G., DA SILVA, D.A., BUGLIONE, L., SCHEIDT, R. & PRIKLADNICKI, R. 2010. Best practice fusion of CMMI-DEVv1.2 (PP, PMC, SAM) and PMBOK 2008. *Information and software technology*, 52:749–757. Available: Science Direct. Date of access: 11 Nov. 2010.
- WEBBER, L. & WEBBER, F. 2009. IT project management essentials. Austin: Aspen. 732 p.
- WELLS, J.D. 2000. Extreme programming project. <http://www.extremeprogramming.org/map/project.html> Date of access: 6 May 2011.
- WESTFALL, L. 2009. The certified software quality engineer handbook. Milwaukee, Wisconsin: ASQ Quality Press. 640 p.
- WESTLAND, J. 2006. The project management life cycle: a complete step-by-step methodology for initiating, planning, executing and closing a project successfully. Philadelphia, PA: Kogan Page. 237 p.
- WESTNEY, R.E. 1992. Computerized management of multiple small projects. Boca Raton, FL: CRC Press. 357 p.
- WIDEMAN, M. 2002. Comparing PRINCE2 with PMBOK. <http://www.maxwideman.com/papers/comparing/comparing.pdf> Date of access: 6 May 2011.
- WILLIAMS, L. 2007. A survey of agile development methodologies. 19 p. <http://agile.csc.ncsu.edu/SEMaterials/AgileMethods.pdf> Date of access: 6 May 2011.

WYNEKOOP, J.L. & RUSSO, N.L. 1993. Systems development methodologies: unanswered questions and the research-practice gap. (In DeGross, J.I., Bostrom, R.P. & Robey, D., eds. Proceedings of the Fourteenth International Conference on Information Systems. Orlando, FL: ACM. p.181–190).

WYNEKOOP, J.L. & RUSSO, N.L. 1997. Studying system development methodologies: an examination of research methods. *Information systems journal*, 7(1):47–65, Jan.

WYSOCKI, R.K. 2011. The business analyst/project manager: a new partnership for managing. Hoboken, NJ: Wiley. 240 p.

YEONG, A. 2009. The marriage proposal of PRINCE2 and PMBOK.
<http://www.anthonyyeong.com/The%20Marriage%20of%20PRINCE2%20and%20PMBOK.pdf>
Date of access: 6 May 2011.

YIN, R.K. 2003. Applications of case study research. Thousand Oaks, CA: SAGE. 173 p.

YOUNG, R.R., BRADY, S.M. & NAGLE, D.C. 2009. How to save a failing project: chaos to control. Vienna, VA: Management Concepts. 234 p.

YOUNG, T.L. 2007. The handbook of project management: a practical guide to effective policies, techniques and processes. Philadelphia, PA: Kogan Page. 295 p.

ZAVAL, L.K. & WAGNER, T. 2009. Project manager street smarts: a real world guide to PMP skills. Hoboken, NJ: Wiley. 381 p.

ZELKOWITZ, M.V. 2004. Advances in computers: advances in software engineering. Amsterdam: Academic Press. 368 p.