Project Management Maturity Integration based on Capability Maturity Model Integration

مشروع تكامل نضج الإدارة القائمة على نموذج نضج القدرات التكاملية

By:

Eng. Taher Ghazal
110071

Dissertation submitted in partial fulfillment of the requirements for the degree of MSc Information Technology Management

Faculty of Engineering and Information Technology

Dissertation Supervisor:
Dr. Khaled Shaalan

2013

This dissertation is submitted as part of a M.Sc. degree in Information Technology Management at British University in Dubai.

ACKNOWLEDGEMENT

I thank the Almighty God for keeping me healthy during the time I was working on this dissertation.

I would like to express the deepest appreciation to Prof Dr. Khaled Shaalan, my supervisor who has the attitude and the substance of a genius: he continually and convincingly conveyed a spirit of adventure in regard to research and an excitement in regard to teaching. Without his guidance and persistent help this dissertation would not have been possible.

I would like to sincere thank Dr. Tariq Rahim Soomro of Al Ain University of Science & Technology, for the support, guidance, encouragement, advice and direction while I was carrying out this research work.

In addition, a thank you to Eng. Ali Mahdi, who stand for me while peer reviewing.

I would also like to thank "European Journal of Scientific Research" ISI Journal that has an impact factor of (0.736) for publishing my paper on this dissertation in Volume 99 Issue 3 under article titled "Integration of Project Management Maturity (PMM) based on Capability Maturity Model Integration (CMMI)".

I would also wish to thank British University in Dubai (BUiD) Staff for providing the opportunity to follow the course and carry out my work in regard of copyrights issues.

Finally I also would like to thank my wife for understanding during the time I was busy working on this dissertation.

مشروع تكامل نضج الإدارة القائمة على نموذج نضج القدرات التكاملية

الملخص

تطوير البرمجيات هي قطاع الصناعة الذي يمثله التغير السريع والتقدم التكراري.

ولكن يعتبر أيضا بأنها الرائدة من حيث الإدارة الفعالة بسبب التكلفة، والموارد والوقت الذي يتطلبه تطوير برمجيات جديدة لحل التحديات القائمة والناشئة. وفي نتيجة وجود عدد من أطر إدارة المشروع في مجال تطوير البرمجيات. وقد وضعت بعض من هذه النماذج القائمة مأخوذة من قطاعات الصناعة الأخرى وظهرت بعضها مع مرور الوقت استجابةً للاحتياجات المتغيرة بسرعة من مطوري البرمجيات وفرق المشاريع الداعمة لها.

هذا البحث دراسة تحديدا لإطارين معينيم والتي تستخدم شعبيا في تطوير الإطار والقدرة على النضج (PMM) البرمجيات، ونضج إدارة المشاريع الإطار. تعتبر السابق الجوانب أكثر عمومية (CMMI) نموذج التكامل من إدارة المشاريع الناجحة ويعتمد اعتمادا كبيرا على الأدب الحاليين يقيم خطيرة القدرة التقنية والتطور CMMI إدارة المشاريع، في حين أن من البرنامج وعملية إدارة المشاريع. هذا البحث يقارن خطيرة ويتناقض اثنين، ويقترح إطارا توليفها التي تستمد أفضل العناصر من كل من هذه النماذج لإنشاء إطار رشيقة التي تشجع أفضل الممارسات والدقة التقنية في مشاريع البرمجيات. تقرر أن إطار توليفها يقلل من المخاطر الكامنة في إدارة المشاريع، ويمكن نشرها بشكل فعال في مجموعة من الحالات.

Project Management Maturity Integration based on Capability Maturity Model Integration

Abstract

Software development is an industry sector which is typified by rapid change and iterative progress. However it is also considered as being leading edge in terms of effective project management because of the cost, resource and time involved in developing new software to resolve existing and emerging challenges. In consequence a number of project management frameworks exist within the field of software development. Some of these have been developed from existing models taken from other industry sectors and some have emerged over time in response to the rapidly changing needs of software developers and their supporting project teams. This research specifically examined two particular frameworks which are popularly used within software development, the Project Management Maturity (PMM) framework and the Capability Maturity Model Integration (CMMI) framework. The former considers the more generic aspects of successful project management and draws heavily upon existing project management literature, whereas the CMMI critically assesses the technical capability and sophistication of the software and the project management process. This research critically compares and contrasts the two and proposes a synthesised framework which draws upon the best elements of both of these models to create an agile framework which encourages best practice and technical precision within software projects. It is determined that a synthesised framework reduces the risks inherent in project management and can be effectively deployed in a range of situations.

Keywords: *Project Management; Software Projects; PMM; CMMI*

Word count = 10,542 (not inc acknowledgement abstract, table of contents, references or appendices)

TABLE OF CONTENTS

List of Acronyms

CMMI - Capability Maturity Model Integration
EDI - Electronic Data Interface
ERP - Enterprise Resource Planning
OO - Object Orientation
PMBOK - Project Management Body of Knowledge
PMI - Project Management Institute
PMMI - Project Management Maturity Integration
RUP - Rational Unified Process
SDLC - Systems Development Life-Cycle
SEI - Software Engineering Institute
SME - Small to Medium Enterprise
UI - User Interface
UML - Unified Modelling Language

List of Figures

Chapter One

1.0 Introduction

1.1 Contextual Background: Best Practice in Project Management

The Project Management Institute (PMI) was founded in the 1970's with the aim of creating a cohesive body of knowledge relating to professional project management (PMI, 2013). Over time the PMI has come to be regarded as one of the leading authorities on best practice in project management, and at the time of writing the Project Management Body of Knowledge PMBOK as published by the PMI has come is considered to be one of the leading authorities on best practice in project management. Current thinking in the PMBOK asserts that successful project management embraces nine discrete areas of management and control, beginning with the three fundamental aspects of project management scope, time, and quality, and then moving on to embrace wider areas of project management control including communication, risk management and integration. Much of the thinking on basic project management is grounded in the five stage model illustrated in ***figure 1*** below. It is asserted by Burke (2010) that adherence to this model provides the fundamental framework for project management success.

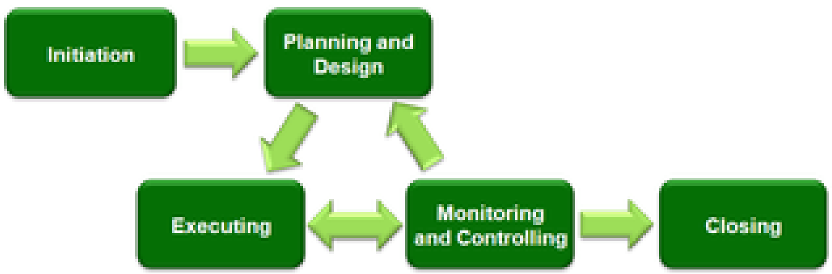


Figure 1: *The Fundamental Framework of Project Management (Source: adapted from Burke, 2010)*

There are, however, other professional bodies which also devote themselves to the school of thought pertaining to successful project management. In the late 1980's the US Department of defence (DoD) sponsored work at Carnegie Mellon University focusing on software engineering as an emerging discipline (SEI, 2013). Collaboratively they developed a tool known as the Capability Maturity Model (CMM) which facilitates baseline assessment of organisational performance within software or IT organisations. Over time it has come to be regarded as the *de facto* standard within the industry. In 2000 the model was renamed the Capability Maturity Model Integration (CMMI) on the basis that a very large proportion of the work conducted by software and IT organisations relates to the implementation of new or upgraded systems, and this requires a distinct form of project management. It is asserted by Meskendahl (2010) that over the last decade, the CMMI has come to be regarded as the benchmark standard tool for this process.

The aim of this study is to bring together these two aspects and to identify how they can be integrated to form one unified tool which can be applied in a wider range of software project management situations. Currently, there remains alarmingly high number of software project failures, with the term "failure" being used as an umbrella term to cover a range of distinct issues (Levin, 2012). It is posited that use of a hybrid tool drawing upon best practice from the PMBOK and the CMMI will create an all-embracing framework which will allow organisations to identify potential issues before they become problems, thereby averting project management failure in a software context. Moreover it is hoped that the use of an integrated framework will increase the efficacy of project management and will allow the stakeholders involved to have greater control and visibility of any potential issues. It is the intention to use existing literature to form a holistic picture of these frameworks in application, and thus to identify how best to amalgamate the two schools of thought for future use. (Taher Ghazal, 2013)

1.2 Research Aim and Objectives

The principal aim of this research is to critically assess the literature relating to best practice in project management software environment, and then to synthesise the PMM and CMMI to develop a conceptual framework which is fit for purpose in the contemporary business environment. It is the intention to undertake this exercise as a literature-based review, theoretically tested against existing case study data and practitioner evidence where it is available. To achieve this outcome the following objectives are proposed:

- A critical assessment of the benefits and challenges of PMM, drawing upon existing theoretical and empirical evidence.
- A critical assessment of the benefits and challenges of CMMI, also drawing upon theoretical evidence and empirical knowledge.
- A synthesis of the two frameworks and subsequent analysis to create a new conceptual framework suitable for future project management software environment, with potential application for wider project management use.

It is the contention of this research that this represents a fresh avenue of study which will augment the existing body of knowledge and help both academics and practitioners to enhance and improve approaches to project management in an agile and dynamic environment. (Taher Ghazal, 2013)

1.3 Project Management in the Software Sector

Bocij *et al* (2008) indicate that project management within the software sector represents overall best practice in terms of project management as a discipline. This is because the resource investment and agility required to successfully managing projects with a significant number of variables and unknowns are inherent characteristics of the software sector. Bespoke and innovative software systems are deliberately designed to differentiate themselves from existing solutions, meaning that it is incumbent upon software providers and developers to scope out and quote for new systems which allow businesses to differentiate themselves. The ability to assess likely resource requirements and costs for such projects require a specific approach to project management, which is supported by ongoing best practice as the project itself goes live (Holcombe, 2008). Therefore not only must software developers have a high degree of technical capability, there must also be capable of managing complex projects, which may potentially be running in multiple streams (Atkinson *et al,* 2006). Consequently software developers and project managers in the software sector have come to be regarded as some of the leading experts in the field of project management at a global level. This also leads to the situation where project managers from software environment are asked to transfer their skills to other industry sectors to help improve best practice in project management.

1.4 Dissertation Structure

This dissertation will be divided into five chapters to facilitate a logical framework in the research and subsequent discussion. The first chapter has provided the background, aim and objectives of the research. The second chapter provides a critical discussion of the relevant literature and empirical evidence. The third chapter sets out the research method and justifies the use of secondary study and literature review as the preferred research approach. The fourth chapter develops the synthesised conceptual framework and critically considers the anticipated benefits and likely challenges. Finally, the last chapter presents the conclusions and recommendations, and identifies areas for further research. (Taher Ghazal, 2013)

Chapter Two

2.0 Literature Review

2.1 Introduction

This chapter presents a critical analysis of existing literature relating to project management in a software environment, the PMM and the CMMI. It also considers the benefits and challenges associated with project management, the use of these conceptual frameworks, and the likely implications of synthesis in a software environment and in the wider body of knowledge relating to project management in general. It is the intention in this chapter to present a balanced view of these conceptual frameworks, and to illustrate how they were developed and how they could be improved for future use. Therefore, this chapter will begin with a definition of project management, and will then critically examine the models drawing upon a range of work from academics and practitioners who have used the models to present a holistic view of these models in action.

2.1.1 Definition of Terms: Project Management

According to Burke (2010:17) at its most basic level project management can be defined as:

"*The discipline of planning, organizing, motivating, and controlling resources to achieve specific goals*"

This view is shared by the PMBOK (2012), who expand upon this definition and suggest that because projects by their very nature are discrete endeavours with a defined timescale, there is a need to acknowledge the constraints of time as part of the definition of project management. Further to this Maylor (2010:23) suggests that good project management is concerned with the introduction or demonstration of value in exchange for resource and effort, and therefore these should also be acknowledged in the definition of best practice in project management. He suggests that this could potentially take the form of benchmarking or retrospective assessment and evaluation to determine the extent to which a project has achieved its objectives. As this is also part of the basic framework of project management expressed in chapter 1 above, then it is suggested that a combined definition drawing upon the suggestions of the PMBOK (2012) and Maylor (*ibid*) should form the foundation of the definition of project management.

Further to these discussions Bocij *et al* (2008:41) suggest that within the software sector, it is necessary to acknowledge the idiosyncrasies of this industry, and therefore definitions of project management should incorporate the specifics relating to software project management. To this effect the *Systems Development Life-Cycle* (SDLC) is defined as:

"*The process of developing information systems through investigation, analysis, design, implementation and maintenance*" (Marakas and O'Brien, 2011:19)

According to Post and Anderson (2006:24), the SDLC represents a distinct niche of project management which is concerned with the tools and techniques associated with software development and project management, including *inter alia*, agile methodologies, the waterfall method, prototyping and incremental development. ***Figure 2*** below provides an illustration of how the SDLC functions as an evolutionary cycle.

Figure 2: *The Systems Development Life-Cycle (Source: Adapted from Post and Anderson, 2006)*

This definition has application within this research because it is the foundation of both the PMM and the CMMI in terms of iterative development of software project management to provide the best possible solution for end users.

2.2 Project Management Frameworks

2.2.1 Project Management Maturity (PMM)

The Project Management Institute (PMI, 2013) has proposed the following definition of project management maturity (PMM):

"*Organisational Project Management Maturity describes an organisation's overall ability to select and manage projects in a way that supports its strategic goals*"

Further to this they developed a checklist to help an organisation or project team assess the relative maturity of their approach to project management. A copy of this is contained in ***Appendix 1***. Over time the PMI have formulated a well-defined project management structure which can be adapted and applied in a range of industry sectors including software development and information technology, construction, engineering. According to the PMI, project management embraces nine discrete areas of knowledge which are as follows:

1. Scope
2. Time
3. Cost
4. Quality
5. Human Resource
6. Communication
7. Risk
8. Procurement
9. Integration

The PMI assert that a good project manager or project team should be capable of functioning at a superior level in all of these areas to comprehensively complete a project to a high standard. As a result, good project managers are expected to work multiple "hats" throughout the life-cycle of a project so that they can not only address the needs of team members and wider stakeholders, but also manage the technical and budgetary aspects of a large-scale, high-value and potentially high risk project (Atkinson *et al*, 2006).

It is acknowledged in practitioner discussions and literature that the PMM approach is resource intensive and is also reliant upon the project team or the project manager having considerable expertise within a particular area in order to facilitate effective project management (Kearns and Lederer, 2003). Furthermore there is acceptance of the fact that in order to overcome the known challenges of project management, particularly issues such as increased risk (Wang and Yuan, 2011) and problems with "scope creep" (Burke, 2010; Atkinson, 1999), it is imperative to maintain tight control of the project and to respond rapidly to any potential problems which may cause the project to miss objectives or targets. This is why the PMI firmly advocate the use of the PMM framework to assist project teams in monitoring the performance of the project in the nine areas listed above. Further to this, Lee and Pai (2013) assert that application of the PMM delivers several benefits including (i) the ability to articulate project success in a manner that external stakeholders can appreciate; (ii) the ability to objectively measure project performance against targeted aims; (iii) increase the predictability of project delivery and the accuracy of project timescales and costs; and (iv) encourage internal communication in a multi-project environment to increase the effectiveness of resources, and potentially contract the overall timescale without adversely impacting on quality. For these reasons the PMM has come to be acknowledged within the software industry as a comprehensive and versatile conceptual framework which can be applied in a range of circumstances (Luftman, 2000).

With regard to assessing the maturity of project management Cooke-Davies and Arzymanow (2012) believe that there are over 35 models which can be used to assess the maturity and capability of project management teams and organisations to deliver a project. The core distinction is between technical capability and understanding the importance of effective project management in order to maximise resource utility. Killen and Hunt (2013) explains that knowing the difference between effective project management and effective program management for multiple software projects is a useful acid test of the maturity of project management capability and integration within organisations. Kerzner (2007) describes this as the *Project Management Maturity Integration* (PMMI) framework. Kwak and Ibbs (2002:152) acknowledge that despite the difficulties associated with PMMI, not least of which the fact that it requires open and honest assessment of capability, is that it can be a "painful" process for the stakeholders concerned as in order to move forward it is necessary to accurately identify inhibitors and challenges within the project which caused resource inefficiency. This leads on to the use of the *Capability Maturity Model Integration* (CMMI) which is considered by contemporary practitioners to be a favoured model for software projects in particular (Chouhan and Mathur, 2012; Popovic *et al*, 2010).

2.2.2 Capability Maturity Model Integration (CMMI)

In its most direct form the capability maturity model (CMM) is a framework which helps organisations to increase process efficiency across a range of business operations (Chouhan and Mathur, 2012). The CMM framework was first presented by Paulk *et al* (1993) at Carnegie Mellon University in response to evolution in software development and project management. The framework has subsequently been adopted in other industry sectors to illustrate how improvements in business processes can significantly increase organisational efficiency (Popovic *et al*, 2010). The CMM shares aspects of best practice project management and lean techniques to facilitate an iterative approach the process improvement which results in increased efficiency and effectiveness (Cegielski *et al*, 2005). Holcombe (2008:137) explains that under this approach business processes are rated or measured according to their level of maturity using the following five benchmarks:

1. Initial;
2. Repeatable;
3. Defined;
4. Quantitatively Managed;
5. Optimizing

Figure 3 below provides a visual illustration of how the five stages build upon one another:

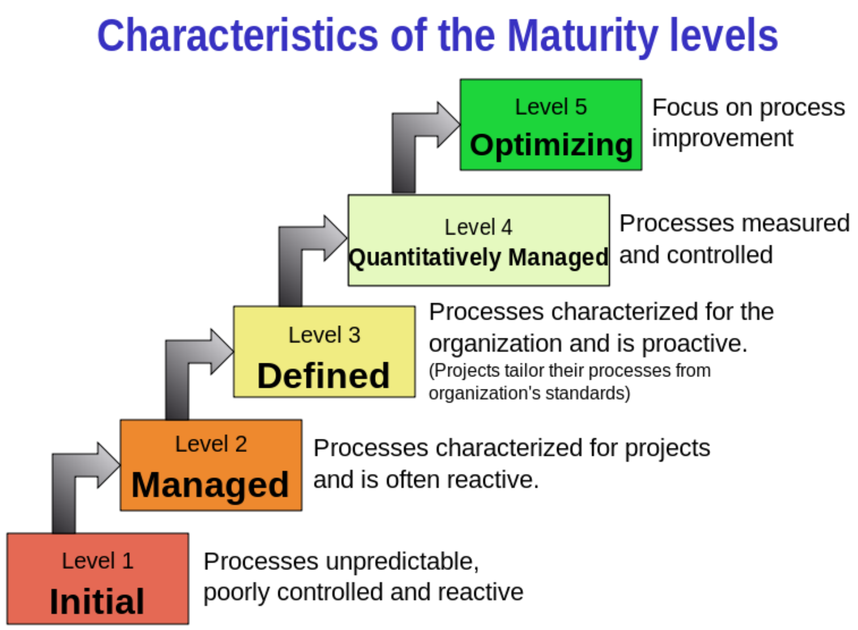


Figure 3: *Characteristics of Maturity Levels within CMM (Source: adapted from Paulk et al., 1993)*

Englund and Graham (1999) explain that the CMM was originally developed to synthesise a number of different benchmark approaches and conceptual frameworks within the field of software development in order to provide a cohesive framework in the industry which could be used on an international basis. The fundamental aim of the CMM is to help an organisation align all of its business processes to achieve and support overall organisational objectives. Gardiner (2005) expands that the CMM has been superseded by V1.2 known as the *Capability Maturity Model Integration* (CMMI) which looks at best practice when integrating and implementing new business processes on a cross organisational basis. Obvious examples include the use of systems to transfer data between organisations on a real-time basis in order to improve organisational efficiency. These may include Electronic Data Interface (EDI) and Enterprise Resource Planning (ERP) systems (Chaffey, 2009; Bocij *et al*, 2008).

According to Trieu and Jose (2010) the CMMI has two forms of operation; (i) continuous; and (ii) staged or incremental. As these terms suggest, continuous application refers the development and maintenance of the status quo and incremental refers to the ongoing development and process improvement to deliver organisational efficiency. Further to this Meskendahl (2010) explains that the CMMI operates in three fundamental areas (i) acquisition; (ii) services; and (iii) development. This is because these three areas essentially describe overall business operations at generic level. From a practical perspective organisations are appraised as to their maturity level and a score is awarded. Effectively this benchmarks the organisation and they can use this knowledge to improve their process maturity and thus by extension the efficiency and integration effectiveness (Williams *et al*, 2012). Post and Anderson (2006) argue that within large-scale software development understanding the level of maturity can help increase the level of accuracy and control and project management and software development to improve profitability and customer satisfaction. Interestingly, Turner and Jain (2002) assert that there is a broad area of overlap between CMMI and lean techniques because both rely on iterative process development in order to improve organisational performance on a perpetual basis. Further to this it is suggested by Jensen (2012) that the linkage between CMMI and lean helps to transfer best practice knowledge the other industry sectors outside of software development.

The main challenges with using the CMMI relate to the very rigid approach to process control which can in some instances be overly oppressive, especially in agile project development (Basu *et al*, 2003) or small-scale operations (Benamati and Lederer, 2000). Also, the evolutionary nature of software development and the fact that it is a rapidly changing industry mean that generic application of the CMMI can result in a sub optimal outcome (Cegielski *et al*, 2005). Therefore Chan *et al* (2006) believe that whilst there is a great deal of merit in the CMMI, a degree of bespoke application is in fact preferable to ensure that the organisation retains its unique source of competitive advantage and adheres to the overall organisational aims and objectives.

2.3 Challenges in Project Management

According to Lanvin (2012) all projects are at risk of project failure, although the root causes of project failure are varied and in some cases difficult to control. Considerable literature and research has been devoted to identifying consistent themes in project management failure such that they can be isolated and rectified before they become an issue. Overall it is suggested by Maylor (2010:78) that the most common themes of project failure and challenge can be summarised as follows:

- Poor definition of project management requirements, including issues such as scope, resource requirements, and budget availability
- Failure to manage the expectations of stakeholders involved in a project, both internally and externally, leading to unrealistic demands and ineffective resource management.
- Difficulties in communication, especially in international projects as a result of flawed fundamental assumptions as the expectations and requirements
- Weak project management as a result of a lack of structural framework and/or a lack of project management experience
- Failure to respond effectively to external factors influences which have the potential to derail a project, including problems with external suppliers or extrinsic variables beyond the control of any of the stakeholders

According to Maylor (2010:79) any one of these challenges has the potential to cause project failure, and this is why the application of either the PMM or the CMMI can significantly reduce the risks associated with project management, and enhance the effectiveness and efficiency of the project. However it is noted by Gattiker and Goodhue (2005) that relatively few organisations have the necessary experience to apply these frameworks effectively. Hussin *et al* (2012) posit that one of the main problems is the rigidity of the frameworks and the fact that they require considerable experience and resource to deploy them to maximum effect. Kagioglou *et* *al* (2012) also suggest that the rapid evolution of the software industry has caused some practitioners to overlook these frameworks because they are considered old-fashioned in the contemporary environment. All of these factors present a challenge to the use of these models, and although they have very obvious benefits, there is a line of argument which potentially suggests that synthesis and streamlining of these two approaches would have greater benefit for small organisations and those project teams with less experience and resource. Overall industry as a whole would benefit from increased success rates in project management, and so there is a practical reason for developing a more agile approach to control the project management utilising the best elements of these frameworks.

2.4 Synthesis of Frameworks

There is an emergent body of knowledge which advocates the synthesised use of recognised software development frameworks such as Extreme Programming (Turner and Jain, 2002), Agile (Krzanik *et al*, 2010), Scrum (Diaz *et al*, 2009) and even classical techniques such as the waterfall method (Eberlein and Jiang, 2009). Suffice to say in all of these cases there are arguments both for and against the melding of alternative approaches. Whilst at a holistic level in the software development community there is broad support for the view that a blending of approaches which moves knowledge forward is a good thing, there are also a number of voices of caution such as Santana *et al* (2009) who argue that before it is possible to synergise approaches it is necessary for a slight dilution or adaptation of each perspective or contribution in order to maximise the capability of the new process. They believe that all too frequently there are attempts to bring together two or more alterative methodologies using a "bolt on" approach leading to a sub-optimal solution which fails to deliver a superior performance or result. Abbas *et al* (2008:107) argue that at best current thinking in this area is "ad-hoc" leading to a fragmented outcome which is in some parts superior and some parts inferior. Krzanik *et al* (2010:3) go further than this and suggest that it is time for an entirely new perspective which is not solution driven but value driven and encourages developers to identify sources of value before establishing which aspects and techniques of project management would be required. They also assert that it is necessary to consider wider or multiple perspectives when planning and preparing for project development to ensure that a project is accurately scoped. Both Krzanik *et al* (2010:4) and Santana *et al* (2009:127) believe that despite significant advances in software and project management techniques, a failure to communicate effectively at the outset and scope projects accurately remains as one of the greatest inhibitors of project management success.

2.5 Summary

This chapter has presented the comprehensive definition of project management, an overview of both the PMM and the CMMI, and an identification of the benefits and challenges. There has also been a discussion as to the risks and benefits associated with synthesis of established project management models including *inter alia* agile, scrum, extreme programming, and classical techniques. It is evident from the literature that there is a clear desire within the software community to be as proactive and effective as possible when developing software. However it is also suggested that the perennial problem of poor communication between project management stakeholders remains one of the greatest challenges to project management success. There is also some concern as to the blending of established software development frameworks such that maximum utility is obtained as opposed to suboptimal performance through rigid application of established frameworks. As the overarching aim of this study is to bring together two well-established software development frameworks in the synthesised fashion it will be necessary to consider the benefits and challenges in quite considerable depth. Therefore the following chapter presents a discussion of the methodology which will be used in this study, the results of which will be shown in chapter 4.

Chapter Three

3.0 Methodology

3.1 Introduction

This chapter describes and justifies the research methodology applied in this study. According to Ashley and Orenstein (2005) research studies such as this which blend empirical findings and theoretical considerations in attempts to develop a new conceptual framework are firmly rooted in the sociological school of thought. As such there are a number of alternative research methods and supporting approaches which could be adopted to achieve the overall research aim. Therefore, this chapter critically contrasts the schools of thought as regarding research philosophies and approaches, and also the detail of the research strategy and data collection. This will be followed by a description of the data analysis process utilised in this study.

3.2 Research Design

Bryman and Bell (2011) explain that the research method or design comprises a number of inter-related elements which are built up in a layered or structural fashion to ensure efficacy of research and reliability in the outcome. In consequence, this chapter will also adopt this layered approach, commencing with an assessment of the most suitable research philosophy and a justification of why the chosen strategy and data analysis method is the most apposite. Throughout the course of this chapter there will be regular reference to classical and contemporary thinking in respect of research studies to illustrate why the selected methodology is the most relevant.

3.2.1 Research Philosophy and Approach

Broadly speaking, there are two fundamental and directly opposed schools of thought in regards to research philosophy (Christensen *et al*, 2011). These are the positivist school of thought, and the interpretivist school of thought. Christensen *et al* (2011) explain that the former is typically concerned with empirical, statistical study grounded in large bodies of research which strive to illustrate the nature of relationships between variables within a dataset. For example testing or ascertaining the level of reliability of using a certain project management framework and the subsequent frequency of success in the ensuing projects. Christensen *et al* (2011) also explain that positivist research typically engages the use of hypotheses which are then proved or disproved as a result of the empirical study. At the opposite end of the spectrum Collins and Hussey (2009) describe the research philosophy known as interpretivism, which they explain is typically focused on understanding the perspectives of research participants or stakeholders within a certain subsection of society. It is normal in interpretivist research to focus on the context of the study and the responses of the research participants to their contextual understanding, as this can help to describe and explain their behaviours. For example, within the context of this study using an interpretivist approach and gathering primary data which explains why project management stakeholders adhere to or deviate from accepted best practice techniques might help to explain why a new synthesised approach to project management is relevant.

Zigmund *et al* (2012) reiterate that these contrasting views are simply alternative perspectives to help gain a detailed understanding of the given research situation or phenomena. This being said, depending on the specifics of the research problem it is usually preferable to use one type of research philosophy over the other, unless of course a mixed-methods research approach is called for. As this research is looking to adopt a conceptualised approach which draws upon existing evidence with the intention of developing a synthesised framework, it is proposed that in fact just such a mixed methods approach utilising inductive and deductive logic is required. Bryant (2002) and Charmaz (2006) refer to this is grounded theory. In brief, grounded theory can be considered as a "reverse engineered" version of positivist enquiry, although it draws heavily upon the principles of interpretivist study (Charmaz, 2006). Grounded theory commences with the collection of a wide body of data which is then systematically analysed using principles typically associated with interpretivist enquiry, such as thematic coding and cross-referencing (Zigmund *et al*, 2012). The output of grounded enquiry is a suite of hypotheses which are proved or disproved relative to the body data originally collected. This satisfies the positivist requirement for rigorous scientific study with the proved or disproved of the hypotheses, however it relies upon interpretivist concepts to explain and justify the hypotheses relative to the data. Given that this study is seeking to develop a new synthesised framework this seems to be the most appropriate route to follow.

Critics of grounded theory such as Thomas and James (2006) suggest that it has already diverged from its original construct as proposed by Glasser (1992). For example there is some debate over the best approach to coding, although Charmaz (2006) argues that this is no different from a pure interpretivist study which requires justified albeit subjective coding on the part of the researcher. Thomas and James (2006) also argue that there are some limitations to the approach as regards initial data collection because they suggest that conducting a literature review prior to grounded theory analysis attracts the risk of inadvertent subjectivity. Once again there is a robust rebuttal of this criticism, this time from Corbin and Strauss (2008) who points out that this is no different from the inherent challenges of both interpretivist and positivist study when there is a need to make a decision about what should be included in research on what should be dismissed. They go so far as to suggest that in fact grounded theory blends together the best aspects of both schools of philosophical thought leading to a far more robust philosophical foundation and either of these approaches in isolation. In response to this it is suggested that because of the nature of the research study which is seeking to develop existing contemporary knowledge, the grounded theory approach is the most suitable as this will facilitate extension of existing concepts, tested using a bilateral approach of empirical evidence and conceptual theory. Arguably this will result in a more robust framework which acknowledges the charges discussed in the literature review, such as the risk of rigid adherence to principal at the expense of stakeholder perspective.

3.2.2 Research Strategy and Data Collection

As it has been ascertained that grounded theory is the most suitable philosophical approach to adopt in this research, it is clear that the study will now rely fundamentally upon secondary data which is already available in the public domain. Therefore the next questions to ask is where the data will be collected from and what will determine the parameters of the data. Crotty (2005) suggests that it is necessary to clearly define the parameters of the research in order to illustrate inclusion of the alternative perspectives and also to defend necessary the selection of the data which has been included. Further to this, Saunders *et al* (2012) point out that it is necessary to explain where the data was acquired from in order to justify its inclusion. Accordingly, data for this research study was gathered from reputable and reliable sources including academic and practitioner journals and literature, presentations from conferences and to a limited extent the website of relevant professional bodies including the Project Management Institute (PMI) and the Software Engineering Institute (SEI). Given that this study is seeking to explore contemporary and forward facing aspects of software development it is necessary to rely upon recent data which challenges and critically assesses current thinking (Zigmund *et al,* 2012). For this reason greater emphasis will be placed on evidence presented at conferences and recent literature as this represents recent thinking in a dynamic body of knowledge. There will however be inclusion of recommended best practice in classical thinking to provide a foundation and a comparator for the proposed conceptual framework.

For these reasons the search terms identified during the data collection process included the following:

- Project management maturity (PMM)
- Capability Maturity Model Integration (CMMI)
- Agile Software management
- Project Management
- Extreme programming
- Scrum management

Preference was also given to empirical research studies which demonstrated the efficacy or otherwise of these approaches using case studies and primary evidence (Saunders *et al*, 2012). This was deliberate in order to ensure that sufficient proof was presented of the likely capability of the synthesised framework.

3.2.3 Data Analysis

Under the principles of grounded theory, data analysis took place using a four stage process of coding, cross assessment, conceptualisation and development of theory, in this case the presentation of an augmented and synthesised project management framework. As the aim of this study was to present a new approach bringing together the best elements of the PMM and the CMMI, this required detailed consideration of the respective merits and challenges as elucidated it by previous academics such as Krzanik *et al* (2010) and Santana *et al* (2009). The detail of this data analysis and ensuing conclusions set out in chapter 4 below. The discussions in chapter 4 also highlight the difficulties experienced during the process and acknowledgement of the fact that certain lines of enquiry require further investigation and evidence in order to prove or disprove specific concepts relating to the development of a robust theory. According to Charmaz (2006) this is in fact a normal element of grounded theory which has inherent risk of failing to deliver any innovative outcome. However despite these setbacks it was possible to present a new framework as will be shown in the following chapter.

3.3 Summary

This chapter has described and justified the use of grounded theory in order to satisfy the aim and objectives of this research. There has been discussion of the suitability of grounded theory and acknowledgement of its inherent weaknesses and benefits leading to its selection for the study. There has been a discussion of the differing requirements of grounded theory relative to other approaches and a description of the search terms and specifics of the data analysis leading to the presentation of a new conceptual framework in this area of research. Accordingly the following chapter sets out the process of grounded theory used in this study and presents the new proposed conceptual framework.

Chapter Four

4.0 Findings, Analysis and Critical Discussion

4.1 Introduction

This chapter presents the results, analysis and critical discussion relating to the synthesised framework which is the aim of this research. Under the principles of grounded theory it was first necessary to establish the parameters of inclusion and exclusion from existing evidence leading to the collection of data within these parameters and the subsequent coding, cross-analysis and thematic presentation. Accordingly, this chapter presents the evidence relating to best practice in software management under the principles of PMM and CMMI and also consideration of existing attempts to synthesise software project management frameworks. The evidence and discussions are presented under each of these themes, in conjunction with acknowledgement of difficulties with these discussions such as the fact that in certain areas insufficient evidence was available to prove or disprove any comprehensive lines of enquiry. At the conclusion of this chapter the proposed conceptual framework is presented, which draws upon best practice knowledge and contemporary thinking in this area. It is acknowledged that the proposed framework has some areas which could perhaps be regarded as contentious because of their innovative approach; however it is suggested that these are suitably explained and justified during the course of this chapter following a logical progression of robust analysis of the data.

4.2 Best Practice in Software Project Management

As the aim of this research is to develop future best practice in software project management it seemed prudent to critically assess contemporary recommendations as regards best practice in this area. Further to this there was a need to consider potential "future proofing" of software project management in light of external considerations such as the development of disruptive technology and shifting social perceptions needing to differing stakeholder requirements and expectations (Diaz *et al*, 2009; Boehm, 2002). In light of the discussions of academics such as Krzanik *et al* (2010) and Santana *et al* (2009) it is clear that there are a number of issues to consider in the development of the synthesised framework, not least of which is the inclusion of multiple stakeholder perspectives and acknowledgement of the need for increased flexibility in the interpretation of framework requirements. Therefore, the research in this area focused on the benefits of synthesis and also the challenges highlighting the fact that whilst synthesised frameworks are welcomed within the software development and project management community, there is justifiable reticence in some specific areas. This helps in the development of the new conceptual framework as identification of known weaknesses can be specifically targeted and overcome with the development of a new model. This can arguably be regarded as an innovative contribution to the body of knowledge in this area.

4.2.1 Contemporary Recommendations

There is much in the way of best practice recommendation for successful project management. This has been evidenced in the preceding literature review and also a cursory assessment of any project management textbook, journal or practitioner source. Whilst it has been established that there are basic tenants of best practice in project management which are applicable regardless of the specifics of the project (Meredith and Mantel, 2011; Maylor, 2010; Popovic *et al,* 2010), it has also been established that specific details of the project required alternative approaches (Andersen *et al,* 2006). This has given rise to the development of discrete frameworks such as agile, scrum, and extreme which are specific to the software project management industry. Thus the first stage in the grounded theory process was to establish a common platform contemporary best practice recommendations relating to both software projects in general and more specifically contemporary thinking in software projects as a result of shifting extrinsic circumstances such as disruptive technology. It is the latter element which is particularly interesting as developing a "future proof" solution for such projects in the form of a best practice framework could help to save project teams considerable sums of money and resource in forthcoming projects.

It is accepted in academic research that there are trends in best practice as fresh evidence is discovered and external circumstances change (Beker *et al,* 2009). However, software development appears to possess a specific idiosyncrasy in that acknowledged best practice changes at an accelerated pace (Raber *et al*, 2012; Beker *et al*, 2010), which is almost inevitably a consequence of the speed of development in this field. In consequence the caveat which accompanies this section of the chapter is that this represents best practice at the time of writing, and will almost inevitably be superseded by subsequent research and practitioner development. With this caveat in mind, the following aspects are currently considered as recommended best practice.

Selection of the Development Process: As has been previously alluded to in the literature review, software project management has a number of alternative approaches and according to Crawford (2006) and Chrissis *et al* (2003) selection of the most appropriate approach development process has the capacity to significantly affect the output of the project. This is because this is a fundamental aspect of successful project management and informs all other areas of project software management (Curtis *et al*, 2010). Practitioners and an increasing number of academics recommend selection of the development process is the first step. Interestingly this is an augmentation of general recommended best practice in project management because in most other circumstances the first stage would be generalised planning and scoping (Maylor, 2010). Software projects are different because they can be structured in a different way and selection of the development process is therefore considered by Curtis *et al* (2010) as the most cohesive factor in any software development project. This is because creation of a Rational Unified Process (RUP) means that in practical terms it is much easier to break up the coding into smaller, more easily distributed sections so that the project can be distributed across different resource elements and "crashed" (accelerated) if necessary (DeLone and McLean, 2003). Crawford (2006) asserts that the choice of software development is not as important as consistency and uniformity throughout the process as there is nothing more likely to derail the project than the fact that different aspects have been developed in different ways because this will ultimately require considerable reworking to align resources and create a unified software product.

Resource Requirements: Gable *et al* (2008) believe that resource requirements are the next fundamental aspect necessary in best practice and go on to explain that resource requirements can be further subdivided into (i) functional; and (ii) non-functional. To explain, functional requirements relate to the issues of case for programming and are highly technical nature. Recommended frameworks and guidance exist for this, especially when working on large projects which require coding input from multiple programmers (Humphrey, 1988). Non-functional requirements relate to the characteristics of specifics of the project software output, in common terminology the "look and feel" of the software as it is presented to the user. This can be significantly different from the coding underneath, not least of which is because the vast majority of users need to feel comfortable using the software and therefore it is common practice to present the front screen and user interface (UI) which broadly replicates Microsoft. The simple reason being that the vast majority of users have been brought up with interfaces which have this type of layout and so in order to make them feel at ease it is simply easier to create a front screen which replicates this layout

(Lahrmann *et al*, 2010). From a technical perspective Curtis *et al* (2010) explain that it is usually preferable to identify and plan functional non-functional resource requirements in tandem because they impact on the development of the system architecture, and thus impact on the planning process as well as the extent to which the output of the project meets stakeholder expectation. Arguably this could be considered as part of the planning process although it is highly technical in nature.

Architecture: Matney and Larson (2004) believe that selection of the correct system architecture is absolutely fundamental. This is directly correlated to assessment and identification of resource requirements because of the need to develop a system which not only meets stakeholder requirements but has the capability the future development or "future proofing". Lahrmann *et al,* (2010) point out that from a practical perspective most users will require system development over time because their own business processes and requirements change, and therefore creating system architecture which accommodates these developments will be of benefit in the future. Moreover there is a strong probability that the system developers themselves will create improvements to the system which require new releases and developments and therefore adhering to best practice and system architecture development will make this process much more cost-effective in the future.

Design: Raber *et al* (2012) and Ahern *et al* (2003) believe that good design is fundamental to productivity and profitability in software development projects. Clarke *et al* (2012:3) go so far as to suggest that it is possible to have "good architecture and bad design" by which they mean that the idiosyncrasies of the coding can undermine the development of the architecture and thus that at a future point it will be necessary to rework large sections of the development and design in order to meet future requirements, thereby eradicating all the profitability gains. Curtis *et al* (2010) believe that it is sensible to keep the design as simple as possible, and where necessary utilise OO (Object Orientation) and UML (Unified Modelling Language). This directly links to the work of both Crawford (2006) and DeLone and McLean (2003) who advocate the use of a RUP. It is noted by Negash (2004) that because of the nature of the software development community even on a global basis it is possible to utilise the resources and talent of developers from around the world and still create a unified design, provided that all stakeholders are utilising UML and OO. Service (2009) voices a slight criticism of OO in that he believes it does not necessarily deliver on its functionality promises, and requires considerable resource investment in the first instance in order to create a sufficiently flexible OO. Therefore in commercial terms unless there is sufficient time and resource for this exercise it is usually preferable to focus on using UML.

Code Construction and Testing: From a practical perspective coding often occurs separately to development of architecture because these tasks can be split and resources used more effectively. Clarke *et al* (2012) argue that this creates a risk that the code does not fit within the design and architecture and therefore they suggest either daily testing in order to prevent the need for costly and time-consuming retrospective assessment, or if possible the use of perpetual incremental testing as every piece of code is developed. Known amongst practitioners as "*the daily build and smoke test*" (McConnell, 1996:2), this has been regarded as recommended best practice for nearly 20 years. Although it is a straightforward approach, it has much to commend it because of the fact that it allows developers to identify any issues before they become complex and costly bugs. It is important to reiterate that the smoke test element is an end to end review of the build thus far, with the aim of proving that the system is sufficiently stable to accommodate further build. This is a test of major bugs and not minor resolvable issues, making it a cost-effective use of time and resource.

Performance and Configuration: Tremblay *et al* (2010) suggest that performance testing and configuration are critical aspects of software project management because stakeholders who have paid large sums of money for software expect it to work, and invariably the software will have to integrate and interface with existing legacy systems and potentially other bespoke pieces of software. Wherever possible this configuration testing should take place in a safe environment such as a parallel run which allows users to utilise a system and familiarise themselves with it. This is also an opportunity to test the data can be transferred smoothly between systems in order to deliver the necessary functionality for end users. van Steenbergen *et al* (2008) point out that as this is an obvious and highly visible element of any software project it is usually preferable to ensure that sufficient resources available, not least of which because end users can be nervous about the effectiveness of new systems and resistant to change. Therefore it is imperative to deliver a system which functions effectively in order to ensure that stakeholders are comfortable with the process (Tremblay *et al*, 2010).

Measuring Success: Esterhuizen *et al* (2012) and Clarke *et al* (2012) agree that retrospectively and proactively measuring success against the initial planning and project requirements is sensible, not only from a commercial perspective but also to illustrate the stakeholders the value of what has been produced. This is particularly important for software development which for many end-users does not necessarily reflect the amount of effort and resource which was expended in its development. Therefore clearly defining requirements at the outset and then actively measuring success afterwards helps to reassure stakeholders of the project success and also presents lessons learned for future projects. Bearing in mind the discussions of Curtis *et al* (2010) in respect of using UML and if possible OO, this is also a potential opportunity to reuse or recycle certain aspects of coding in future projects if they have been found to have sufficient generic application. This can help accelerate a project and reduce unnecessary expenditure.

Although some academics recommend a more granular approach to successful project management, the weight of evidence from practitioners illustrates that keeping software development and management projects as simple as possible is preferable for a number of reasons. Therefore the preceding seven stages shown above broadly adhere to generic best practice and project management with acknowledgement of the fact that software development has a number of unique aspects. Moreover, the discussion relating to project management efficiency, and specifically development of code and project management assessment lead on to the next aspect, which is how best to "future proof" the frameworks.

4.2.2 Future-proofing the Frameworks

Very recent discussions amongst practitioners highlight the increased importance of future-proofing software development and to a large extent software project management. Accelerations in technology and increased focus on the cloud have shined a spotlight on software development and stakeholders are increasingly demanding longevity from their very significant software investments (Curtis *et al*, 2010). To a large extent Esterhuizen *et al* (2012) believe that investing the time and resource at the outset in sensible programming and code development is a fundamental aspect of future-proofing, not least of which because it is more effective throughout the life-cycle of the project. However they also acknowledge the importance of regularly reviewing the market to identify fresh developments and technologies which could be synthesised with existing knowledge. Again, one of the unique aspects of the software development community is that they are open to fresh approaches and much more tolerant of change than many other technical disciplines. Evidence of this is found in the work relating to communities of practice and the fact that it is common for software developers to openly share their ideas in order to produce higher quality output (Vierkorn and Friedrich, 2008).

At a recent symposium (Mettler and Rohner, 2012), the theme of the event was "future proofing IT" and much of the work and discussion here centred on the need to unify multiple platforms to create a seamless user experience. Thus contemporary thinking amongst practitioners in this area is the need to focus on bringing together potentially disparate systems so that users have a single experience. Esterhuizen *et al* (2012) pointed out that this has significant cost benefits for clients who are able to utilise remote working and the use of mobile applications for their employees which has cost benefits and helps to reduce carbon impact. Similarly any upgrades and developments to the system can be deployed effectively without the need for multiple applications. Single user licenses across multiple platforms are also a popular area of development. It seems that the theme from the symposium, and indeed from reviewing practitioner blogs, is that it is time to focus on creating a holistic solution whereby IT and software serves the users. To achieve this there must be a greater focus on client led projects which are flexible and adaptive in nature. The practical implications of this are that software developers must be more rigorous and unified in their approach so that they can utilise and leverage their internal knowledge to provide solutions for clients which meet their expectations. The aim is to create a unified platform so that it is not necessary to deploy specialist resource to upgrade across multiple systems. Wixom and Watson (2010) suggest that once this has been achieved greater emphasis can be switched to developing innovative solutions to new problems. However, the key future proofing is to create a seamless user experience for both the developer and the end user.

4.2.3 Benefits of Synthesis

It is interesting to observe that current practitioner thinking already focuses on a synthesised form of the PMM and CMMI, as evidenced by the technical detail in some aspects of current thinking which is more closely aligned with the CMMI, and yet acknowledges the need to interface effectively with all stakeholders which is characteristic of the PMM. What is interesting is that some areas of discussion have already moved beyond both of the PMM and CMMI in isolation, in acknowledgement of the fact that it is now necessary to create a single user experience across multiple platforms for future development. Therefore, synthesising the two frameworks to provide a flexible yet technically focused response which acknowledges the occasionally unpredictable aspect of human interaction has much to commend it. It is suggested that the following benefits can be accrued:

Reduced overall cost: This is quite significant as cost is accrued from multiple stakeholder perspectives. These include the time necessary to manage the project end to end, reduced client expenditure because of single license requirements, use of stable and proven coding under UML and OO best practice which reduces the time spent in testing and any risk of instability, and increased efficiency because of the contracted SDLC.

Simplified Interface: Although from a technical perspective relatively little time is actually expended on the UI because the vast majority of work relates to architecture and coding, the UI is still the most important aspect for end users because this is what they will focus upon. Creating a simplified interface because of a synthesised process which wherever possible utilises existing knowledge and work will help to improve the perception of success on about the client, increasing efficiency and reduce risk. Moreover by focusing on the interface at an earlier stage in the life-cycle process (refer back to the discussions on recommended best practice) it is possible to reshape the architecture and coding in alignment with user needs. Arguably this is a specific form of reviewing stakeholder input throughout the life-cycle of the project thus reducing the risk of scope creep and slippage and ensuring that stakeholder expectations are aligned.

Reduced risk: This is because the technical and human aspects of project management and software development are combined and aligned, which reduces the risk of issues such as scope creep and the need to retrospectively review or assess a project. Reduced risk typically leads to reduce costs and a closer link between the development companies and the end user and client stakeholder also build a stronger long-lasting relationship. It should be recalled that for many pieces of bespoke software purchased by clients there is an expectation of a 5 to 10 year life-cycle of the software *in situ* (DeLone and McLean, 2003) and therefore it is important that there is a good relationship between the software provider and the client stakeholder.

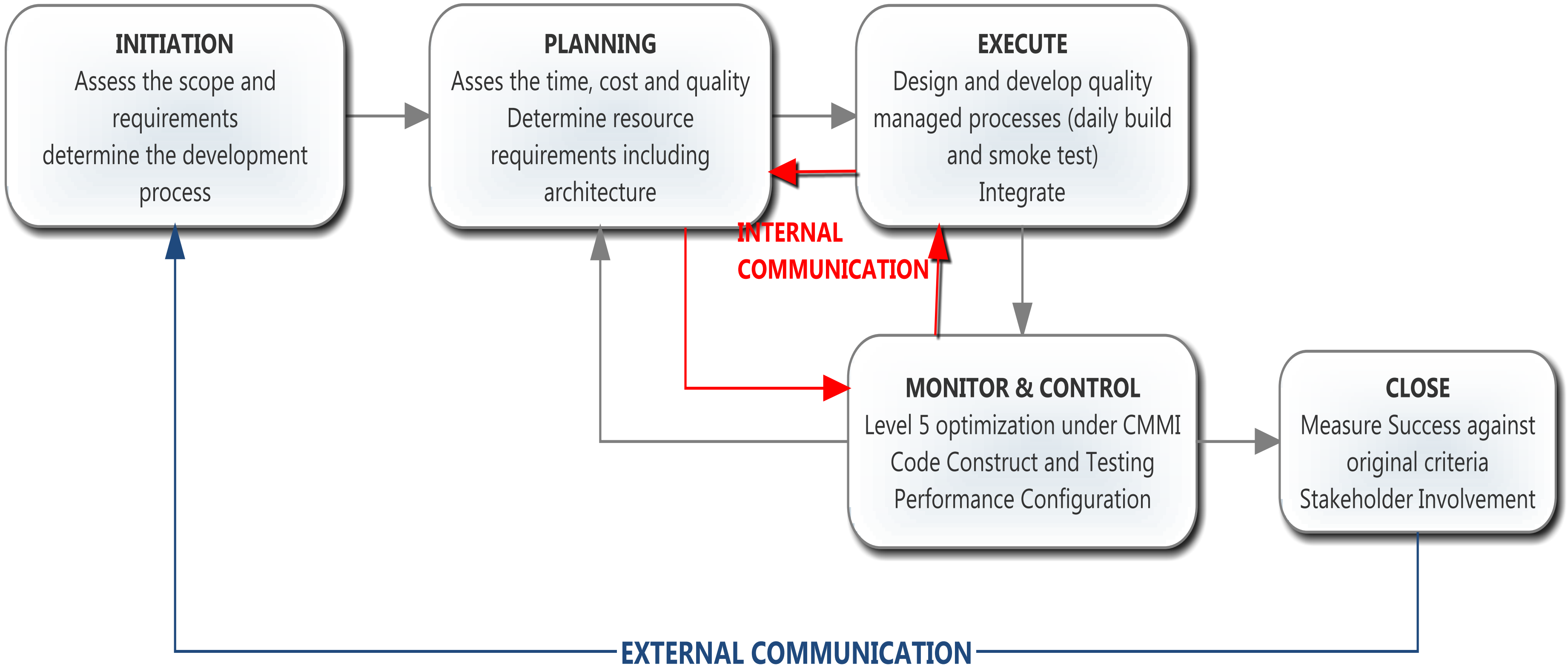
4.2.4 Likely Challenges of Synthesis

It is suggested that there are relatively few potential challenges which may arise from synthesising the two frameworks. Fundamentally because contemporary evidence from practitioners would suggest that they have already begun to adopt a synthesised framework which blends the best aspects of both of these models. Moreover it is noted that for best practice it is necessary at the outset of a project to determine which design framework is best suited, and also that successful deployment of the design framework is far more effective and important than arguing over the granular detail of whether one model has preference to another. In short the world of software development moves too quickly, and so by the time the debate is finished the development will have moved on. For these reasons creating a synthesised model which highlights the importance of technical accuracy with human involvement is likely to be far more effective as a software development tool.

Possible challengers or risks associated with a synthesised model include the fact that some aspects may be inadvertently overlooked; however it is proposed that with a synthesised framework this is a relatively low risk because of the focus on technical accuracy at the outset of the project. The main risks are likely to be associated with obtaining stakeholder buy-in and engagement because of the fact that change is uncomfortable in any project management situation (Crawford, 2006). This is why increased emphasis on the human aspects of project management under the PMM are often more appropriate. There is some concern that technological developments in the wider arena may be inadvertently omitted because of adhering to a defined framework, however with the general trend being a shift towards unified approaches this also seems to be a relatively minor risk which could be retrospectively assessed. One final consideration is the extent to which the first aspects of the project, which are absolutely critical, may be rushed through because they represent relatively little tangible output (Wixom and Watson, 2010). However this is a risk for all projects in the planning stages and so once again it is not considered that this is likely to be a major problem or matter of concern.

Thus, having carefully considered the likely benefits and challenges of the synthesised framework and given due consideration to the prevailing lines of thinking amongst practitioners in the contemporary environment the framework shown in ***figure 4*** overleaf represents the suggested amalgamation of the PMM and CMMI. As the figure illustrates, it has been based upon the foundation of the PMM as a proven framework project management in a generic context, however it draws heavily upon the CMMI because this assesses the level of maturity and integration specifically within software projects. Recalling that the key difficulty is the need to balance robust planning and agility in software projects because of their investigative and developmental nature, the key to flexibility is internal and external communication. This is represented by the red lines shown in the diagram which illustrate the need to have reciprocal communication on an internal and external basis facilitating the flexibility required for project of this nature. (Taher Ghazal, 2013)

Figure 4: *Proposed Synthesised Framework*



4.3 Summary

In summary, this chapter has discussed and critically examined contemporary best practice as preferred by practitioners and advocated by academics, current thinking in regards to future developments in this area, and also the likely benefits and challenges of a synthesised framework. It has been concluded that practitioners are already moving towards the synthesised framework and there is firm recommendation of a unified approach because of the long-term benefits for all the stakeholders involved. It has also been concluded that the likely challenges in respect of the synthesised framework are relatively minimal and largely relate to human aspects such as adapting thinking, as opposed to technical aspects which can be addressed relatively easily through adoption of UML, and where possible OO design which can be applied on a global basis. It seems that there is already demand for a single user interface which can be applied across multiple platforms, and adoption of a synthesised project management framework will help to facilitate this.

Chapter Five

5.0 Conclusions and Recommendations

5.1 Conclusions

To re-cap, the principal aim of this research was to critically assess the literature relating to best practice in project management software environment, and then to synthesise the PMM and CMMI to develop a conceptual framework which is fit for purpose in the contemporary business environment. This involves three principle objectives, *viz*, (i) a critical assessment of the PMM; (ii) a critical assessment of the CMMI; and (iii) creation of a synthesised framework based on contemporary discussions and current best practice to support future developments from multiple stakeholder perspectives. Accordingly, chapter 2 provided a critical consideration of the two existing frameworks and chapter 4 focused on development of the synthesised framework and its likely benefits and risks. Particular consideration was given to the realities of best practice in project management at the time of writing, taking into account the fact that software development is a particularly fast moving environment because of the perpetual innovations in technology and increased demands of stakeholders. Moreover there is recognition of the fact that in many circumstances there is a gap between theoretical best practice and the reality of what occurs for a number of reasons including resource constraints, time considerations and in some instances the technical capabilities of those involved (Wixom and Watson, 2010). The literature review and the practical evidence made it clear that in reality many developers are working under particularly significant pressure to complete projects in a short space of time in order to manage stakeholder expectations, and relatively few have the luxury of being able to use more sophisticated development techniques which cannot be justified in the face of time and cost pressures (Esterhuizen *et al,* 2012).

In conclusion it was established that a synthesised framework would attract relatively low risk in comparison to the benefits that it would accrue. Not least of which is a formalised approach to melding the human and technical aspects of successful software development. There is a clear push amongst practitioners in the wider environment to unify approaches to software development with firm recommendation for UML and wherever possible OO to create a seamless UI. However, in order to achieve this it is necessary to highlight to external stakeholders the need to invest heavily at the outset so that the later stages of the project can be accelerated if necessary and the cost benefits can be achieved. The problem appears to be a gap in understanding and appreciation because of the lack of tangible output in the early stages of software development, and said the use of the synthesised framework which emphasises technical capability and the need to manage external stakeholders is well justified. Moreover software developers themselves are keen to create iterative cycle testing within the life-cycle the project to reduce risk and unnecessary expenditure, however sometimes necessary to enforce discipline to achieve this (Crawford, 2006). Therefore the use of a synthesised framework which can function in a global remit because of the use of UML is considered to be of value. Finally, because the synthesised framework shown at the close of chapter 4 has scope for increased granularity as necessary this recognises the fact that the key to successful project management is not automatically a focus on technical detail at the outset, but rather successful execution and that blending the two approaches should provide a robust outcome. Moreover, the fact that the synthesised framework attracts relatively little risk is a peripheral bonus and not necessarily a key driving factor. (Taher Ghazal, 2013)

5.2 Recommendations

The detailed discussions and analysis which have comprised this study lead the unassailable conclusion that adoption of the synthesised framework would be a benefit for all parties concerned. Not least of which because it helps the software community at large move towards a unified approach which is what end users demand and expect in the current environment. (Taher Ghazal, 2013) One of the idiosyncrasies of software development in comparison to other disciplines and industry sectors is that there is a far more open approach to sharing knowledge and information to help generate the best possible solution (DeLone and McLean, 2003). As a result of this there is also acknowledgement of the fact that recommended best practice moves more quickly because of the perpetual input development in this area. This gave rise to the major change of this project which was to create the framework which could accommodate the necessary technical rigidity required for project management excellence, and the flexibility required for the unique development of bespoke and disruptive solutions to existing problems. Therefore the development of a synthesised framework has helped to accommodate both of these issues. Moreover it should be emphasised that there is nothing to stop use of the synthesised framework at an initial level with subsequent incorporation of specific elements of existing models should further granular detail be required.

Therefore, it is recommended that the synthesised framework is adopted wherever possible however there should be acknowledgement of the fact that it can be adapted as required for the specifics of any project. The only factors which may not be subject to compromise are commitment to technical excellence and project execution, and ensuring that stakeholders are engaged. Therefore this is the other key recommendation when deploying this framework. Other than that the project framework itself is wholly self-explanatory and as it rests upon proven project framework approaches there is little need to reiterate its use. This project has carefully considered the benefits and challenges associated with this and the conclusion has been reached that the risks are no more significant than in any other project, and the application of the synthesised framework helps to reduce them even further. In closing it is considered that this framework can be adopted with immediate effect, and will doubtless benefit from the input and interactions of experienced practitioners and academics in the field.

6.0 References

Ahern, D. M., Clouse, A., and Turner, R., (2003) *CMMI Distilled: A Practical Introduction to Integrated Process Improvement* Boston: Addison-Wesley.

Ashfaque, A., (2012) *Software project management: a process-driven approach* Boca Raton: CRC Press,

Ashley, D., and Orenstein, D.M., (2005) *Sociological theory: Classical statements* (6th Ed) Boston, MA, USA: Pearson Education

Atkinson, R., (1999) Project management: cost, time and quality, two best guesses and a phenomenon, it's time to accept other success criteria, *International Journal of Project Management* Volume 17, Issue 6, December 1999, Pages 337-342

Atkinson, R., Crawford, L., and Ward, S., (2006) Fundamental uncertainties in projects and the scope of project management - *International Journal of Project Management* 24 (2006) 687-698

Basu, V., Hartono, E., Lederer, A.L., and Sethi, V., (2003) The impact of organizational commitment, senior management involvement, and team involvement on strategic information systems planning. *Information and Management* 39(6), 513–524.

Becker, J., (2010) Maturity Models in IS Research", in: Proceedings of the 18th European Conference on Information Systems (ECIS 2010), Pretoria, South Africa 2010.

Becker, J., (2010) Maturity Models in IS Research", in: Proceedings of the 18th European Conference on Information Systems (ECIS 2010), Pretoria, South Africa 2010.

Becker, J., Knackstedt, R., and Pöppelbuß, J., (2009) "Developing Maturity Models for IT Management - A Procedure Model and its Application", *Business & Information Systems Engineering*, 1(3): 213-222

Becker, J., Knackstedt, R., and Pöppelbuß, J., (2009) "Developing Maturity Models for IT Management - A Procedure Model and its Application", *Business & Information Systems Engineering*, 1(3): 213-222

Benamati, J., and Lederer, A.L., (2000) Rapid IT change: nine IT management challenges. *INFOR* 8(4), 336–358.

Benamati, J., and Lederer, A.L., (2000) Rapid IT change: nine IT management challenges. INFOR 8(4), 336-358.

Bocij, P., Greasley, A., and Hickie S., (2008) *Business Information Systems: Technology, Development and Management for the E-Business* (4th edition); Harlow: FT Prentice Hall

Boehm, B. (2002). Some future software engineering opportunities and challenges. *The Future of Software Engineering*, 1-32.

Bryant, A., (2002) ‘Re-grounding grounded theory’, *Journal of Information Technology Theory and Application*, 4(1): 25-42

Bryman, A., and Bell, E., (2011) *Business Research Methods* (3rd Ed) Oxford: Oxford University Press

Burke, R., (2010) *Fundamentals of Project Management: Tools and Techniques* (2nd ed) London: Burke Publishing

Cegielski, C.G., Reithel, B.J., and Rebman, C.M., (2005) Emerging information technologies: developing a timely IT strategy. *Communications of the ACM* 48(8), 113–117.

Chaffey, D., (2009) *E-business and e-commerce management*, (4th Ed) London: FT Prentice Hall p194-200

Chan, Y.,E, Sabherwal, R., and Thatcher, J.B., (2006) Antecedents and outcomes of strategic IS alignment: an empirical investigation. *IEEE Transactions on Engineering Management* 53(1), 27–47

Charmaz, K., (2006) *Constructing Grounded Theory.* London: Sage.

Chouhan, R., and Mathur R., (2012) Role of Software Quality Assurance in Capability Maturity Model Integration *International Journal of Advanced Research in Computer Engineering and Technology* 1(6): 1322-1334

Chouhan, R., and Mathur R., (2012) Role of Software Quality Assurance in Capability Maturity Model Integration International Journal of Advanced Research in Computer Engineering and Technology 1(6): 1322-1334.

Chrissis, M. B., Konrad, M., and Shrum, S., (2003) *CMMI: Guidelines for Process Integration and Product Improvement*, London: Addison-Wesley

Christensen, L. B., Johnson, R. B., and Turner, L. A., (2011) *Research Methods, Design, and Analysis: International Edition*, (11th Ed) London: Pearson Higher Education

Clarke, J.A., Nelson, K. J., and Stoodley, I.D., (2012) The maturity model concept as framework for assessing the capability of higher education institutions to address student engagement, success and retention: new horizon or false dawn? In *First Year in Higher Education Conference 2012*, 26 - 29 June 2012, Sofitel Brisbane Central, Brisbane, QLD.

Collier, K., (2012) *Agile analytics: a value-driven approach to business intelligence and data warehousing* Upper Saddle River, NJ : Addison-Wesley

Collins, C., and Hussey, J., (2009) *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. (3rd Ed) Thousand Oaks, California: Sage Publications

Cooke-Davies, T.J., and Arzymanow, A., (2012) The maturity of project management in different industries: An investigation into variations between project management models international Journal of project management 21(6) 471-478

Cooke-Davies, T.J., and Arzymanow, A., (2012) The maturity of project management in different industries: An investigation into variations between project management models *international Journal of project management* 21(6) 471-478

Cooper, R.G., Edgett, S.J., and Kleinschmidt, E.J., (2009) New product portfolio management: practices and performance. *The Journal of Product Innovation Management* 16 (4), 333–351.

Corbin, J., and Strauss, A.L., (2008) *"Basics of Qualitative Research: Grounded Theory Procedures and Techniques".* (3rd Ed) London: Sage

Crawford, J. K., (2006) The Project Management Maturity Model", *Information Systems Management*, 23(4): 50-58.

Crotty, M., (2005) *The Foundations of Social Research: Meaning and Perspective in the Research Process* London: Sage

Curtis, B., Hefley, W. E., and Miller, S. A., (2010) *The People Capability Maturity Model - Guidelines for Improving the Workforce,* Boston: Addison-Wesley

DeLone, W. H., and McLean, E. R., (2003)The DeLone and McLean Model of Information Systems Success - A Ten- Year Update", *Journal Of Management Information Systems*, 19(4):9-30

Diaz, J., Garbajosa, J., & Calvo-Manzano, J. A. (2009). Mapping CMMI level 2 to Scrum practices: An experience report. *Software Process Improvement*, 93-104.

Englund, R.L., and Graham, R.J. (1999) From experience: linking projects to strategy, *Journal of Product Innovation Management*, 16(1):106-116

Esterhuizen, D., Schutte, C., and Du Toit, A., (2012)A knowledge management framework to grow innovation capability maturity *South African Journal of Information Management* 14(1):1-10.

Forrester, E.C., Buteau, B.L., and Shrum, S., (2011) *CMMI for services guidelines for superior service* Upper Saddle River, NJ: Addison-Wesley

Gable, G. G., Sedera, D., and Chan, T., (2008) "Reconceptualizing Information System Success: The IS-Impact Measurement Model" *Journal Of The Association For Information Systems*, 9(7):377-408

Gardiner, P.D., (2005) *Project Management, a Strategic Planning Approach*, Basingstoke, Palgrave Macmillan

Gattiker, T.F., and Goodhue, D.L., (2005) What happens after ERP implementation: understanding the impact of inter-dependence and differentiation on plant-level outcomes. *MIS Quarterly* 29(3), 559–585.

Glaser, B. (1992). Basics of grounded theory analysis. Mill Valley, CA: Sociology Press.

Holcombe, M., (2008) *Running An Agile Software Development Project*, New Jersey: John Wiley and Sons

Holtsneider, B., (2010) *Agile development & business goals: the six week solution* Burlington, MA : Morgan Kaufmann Publishers/Elsevier

Humphrey, W. S., (1988) Characterizing the Software Process: A Maturity Framework", *IEEE Software*, 5(2): 73-79

Hussin, H., King, M., and Cragg, P., (2012) IT alignment in small firms *European Journal of Information Systems* 11(2), 108–127

Jensen, M.C., (2012) The Modern Industrial Revolution, Exit, and the Failure of Internal Control Systems *The Journal of Finance* Volume 48, Issue 3, pages 831–880, July 2012

Jiang, L., & Eberlein, A. (2009, October). An analysis of the history of classical software development and agile development. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on* (pp. 3733-3738). IEEE.

Kagioglou, M., Irani, Z., and Sharp, J.M., (2012) Information system flow models for new product development processes: speed and flexibility *vs* focus and control. *International Journal of Information Technology and Management* 1(2–3), 168–190.

Kearns, G.S., and Lederer, A.J., (2003) A resource-based view of strategic IT alignment: how knowledge sharing creates competitive advantage. *Decision Sciences* 34(1), 1–29

Kerzner, H., (2007) *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*, (8th Edition) London: John Wiley & Sons

Killen, C.P., and Hunt, R.A., (2013) Robust project portfolio management : capability evolution and maturity International journal of managing projects in business, Vol. 6, Issue 1, (2013), p.131-151

Krzanik, L., Rodriguez, P., Simila, J., Kuvaja, P., & Rohunen, A. (2010, January). Exploring the Transient Nature of Agile Project Management Practices. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on* (pp. 1-8). IEEE.

Kwak, Y., and Ibbs, C., (2002). "Project Management Process Maturity (PM) 2 Model." J. Manage. Eng., 18(3), 150-155. doi: 10.1061/(ASCE)0742-597X(2002)18:3(150)

Kwak, Y., and Ibbs, C., (2002). ”Project Management Process Maturity (PM) 2 Model.” *J. Manage. Eng.*, 18(3), 150–155. doi: 10.1061/(ASCE)0742-597X(2002)18:3(150)

Lahrmann, G., (2010) Business Intelligence Maturity Models: An Overview", in: Proceedings of the VII Conference of the Italian Chapter of AIS (itAIS 2010), D'Atri, Ferrara, George, Spagnoletti (eds.), Naples, Italy 2010.

Lanvin, G. (2012), Rescue the Problem Project: A Complete Guide to Identifying, Preventing, and Recovering from Project Failure. Project Management Journal, 43: 76. doi: 10.1002/pmj.21271

Lee, G.G., and Pai, J.C., (2013) Effects of organizational context and inter-group behaviour on the success of strategic information systems planning: an empirical study. *Behaviour & Information Technology* 22(4), 263–280

Levin, G., (2012), Rescue the Problem Project: A Complete Guide to Identifying, Preventing, and Recovering from Project Failure. *Project Management Journal*, 43: 76. doi: 10.1002/pmj.21271

Luftman, J., (2000) Assessing business-IT alignment maturity. *Communications of the Association for Information Systems* 4(14), 1–50.

Marakas, S., and O'Brien, K., (2011). *Management Information Systems*. New York, NY: McGraw-Hill/Irwin. pp. 485–489

Matney, D., and Larson, D., (2004) The Four Components of BI Governance", *Business Intelligence Journal* 9(3):29-36.

Maylor, H., (2010) *Project Management*, (5th Ed) London: Pearson

McConnell, S., (1996) Daily Build and Smoke Test *IEEE Software*, Vol. 13, No. 4, July 1996

Meredith, J. R., and Mantel, S. J., (2011) *Project Management: A Managerial Approach* (6th edition) New York; Chichester: Wiley

Meskendahl, S., (2010) The influence of business strategy on project portfolio management and its success — A conceptual framework *International Journal of Project Management* Volume 28, Issue 8, December 2010, Pages 807–817

Mettler, T., and Rohner, R., (2009) Situational Maturity Models as Instrumental Artifacts for Organizational Design", in: Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology (DESRIST 2009), Philadelphia, PA

Mutafelija, B., and Stromberg, H., (2009) *Process improvement with CMMI v1.2 and ISO standards* Boca Raton: CRC Press

Negash, S., (2004) Business Intelligence", *Communications Of The Association For Information Systems*, 13(1): 177-195

Paulk, M.C., Weber, C.V., Garcia, S.M., Chirssis, M.B.C., and Bush, M., (1993) *Capability Maturity Model for Software, Version 1.1*. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

Persse, J., (2007) *Project management success with CMMI: seven CMMI process areas* Upper Saddle River, NJ: Prentice Hall

Phillips, J., (2012) *Project management for small business: a streamlined approach from planning to completion* New York: American Management Association

PMBOK Guide (2012) *A Guide to the Project Management Body of Knowledge*, (4th Ed) Bristol: PMI Publishers

Popovic, A., Turk, T., and Jaklic, J. (2010) Conceptual model of business value of business intelligence systems *Management: Journal of Contemporary Management Issues*, (15:1): 5-30

Post, G., and Anderson, D., (2006). *Management information systems: Solving business problems with information technology*. (4th ed.). New York: McGraw-Hill Irwin.

Project Management Institute (PMI) (2013) *About Us* [online] available at http://www.pmi.org/About-Us.aspx retrieved 2nd Feb 2013

Raber, D., Winter, R., and Wortmann, F., (2012) Using Quantitative Analyses to Construct a Capability Maturity Model for Business Intelligence System Science (HICSS), 2012 45th Hawaii International Conference on 4-7th Jan 2012 4219 – 4228

Santana, C., Gusmão, C., Soares, L., Pinheiro, C., Maciel, T., Vasconcelos, A., & Rouiller, A. (2009). Agile software development and cmmi: What we do not know about dancing with elephants. *Agile Processes in Software Engineering and Extreme Programming*, 124-129.

Saunders, M., Lewis, P., and Thornhill, A., (2012) *Research Methods for Business Students* (6^{th} Ed) London: Financial Times Prentice Hall

Services, I. B. M., (2009) *"The New Voice of the CIO", Technical Report*, IBM Global Services, NY: Somers

Software Engineering Institute (SEI) (2013) *About Us* [online] available at http://www.sei.cmu.edu/about/statisticshistory.cfm retrieved 2nd Feb 2013

Springer, M.L., (2010) *Project and program management: a competency-based approach* West Lafayette: Ichor Business Books/Purdue University Press

Tremblay, M. C., Hevner, A. R., and Berndt, D. J., (2010)Focus Groups for Artifact Refinement and Evaluation in Design Research", *Communications Of The Association For Information Systems*, 26(27): 599-618

Trieu, V.H., and Joze, K., (2010) Change Management Strategies for the Successful Implementation of Enterprise Resource Planning Systems *Second International Conference on Knowledge and Systems Engineering* [online] pp178-182

Turner, R., and Jain, A., (2002) Agile Meets CMMI: Culture Clash or Common Cause? *Conference of Extreme Programming - XP Universe* pp 153-165

van Steenbergen, M., (2008)The design of focus area maturity models", in: Proceedings of, St. Gallen, Switzerland pp. 317-332

Vierkorn, S., and Friedrich, D., (2008) *Organization of Business Intelligence",* Technical Report, BARC Institute, Würzburg, 2008

Wang, C., and Yuan, H., (2011) Factors affecting contractors' risk attitudes in construction projects: Case study from China, *International Journal of Project Management,* 29(2):209-219

Williams, T., Klakegg, J.O., Walker, D.H.T., Andersen, B., and Magnussen, M.O. (2012), Identifying and Acting on Early Warning Signs in Complex Projects *Project Management Journal*, 43: 37-53.

Wixom, B. H., and Watson, H. J., (2010) The BI-Based Organization", *International Journal of Business Intelligence Research,* 1(1): 13-28

Zigmund, W.G., Babin, B.J., Carr, J.C., and Griffin, M., (2012) *Business Research Methods* (Intl Ed) Andover: Cengage Learning

Taher Ghazal, Tariq Rahim Soomro and Khaled Shaalan. (2013) 'Integration of Project Management Maturity (PMM) Based on Capability Maturity Model Integration (CMMI)', *European Journal of Scientific Research*, vol. 99, no. 3, April, pp. 418 - 428

7.0 Appendices

7.1 Test of Project Management Maturity

Level	**Process**	**Description**
0	None	The organisation has no project and/or programme management skills or experience.
1	Awareness	The organisation is able to recognise projects and/or programmes, but has little structured approach to dealing with them.
2	Repeatable	There may be areas that are beginning to use standard approaches to projects and/or programmes but there is no consistency of approach across the organisation.
3	Defined	There will be a consistent set of standards being used across the organisation with clear process ownership.
4	Managed	The organisation monitors and measures its process efficiency, with active interventions to improve the way it delivers based largely on evidence or performance based information.
5	Optimised	The organisation will be focussing on optimisation of its quantitatively managed processes to take into account changing business needs and external factors.